Article

A Process for Monitoring the Impact of Architecture Principles on Sustainability: An Industrial Case Study

Markus Funke [1,*], Patricia Lago [1], Roberto Verdecchia [1,2] and Roel Donker [3]

[1] Vrije Universiteit Amsterdam, The Netherlands; {m.t.funke | p.lago}@vu.nl
[2] Univeirsity of Florence, Italy; roberto.verdecchia@unifi.it
[3] Schiphol Group, The Netherlands; roel.donker@schiphol.nl
* Correspondence: m.t.funke@vu.nl

Abstract: Architecture principles affect a software system holistically. Given their alignment with a business strategy, they should be incorporated within the validation process covering aspects of sustainability. However, current research discusses the influence of architecture principles on sustainability in a limited context. Our objective is to introduce a reusable process for monitoring and evaluating the impact of architecture principles on sustainability from a software architecture perspective. We seek to demonstrate the application of such a process in professional practice. A qualitative case study was conducted in the context of a Dutch airport management company. Data collection involved the case analysis and the execution of two rounds of expert interviews. We (i) identified a set of case-related Key Performance Indicators, (ii) utilized commonly accepted measurement tools, and (iii) employed graphical representations in form of spider charts to monitor the sustainability impact. The real-world observations were evaluated through a concluding focus group. Our findings indicate that architecture principles are a feasible mechanism to address sustainability across all different architecture layers within the enterprise. The experts considered the sustainability analysis valuable in guiding the software architecture process towards sustainability. With the emphasis on principles, we facilitate industry adoption by embedding sustainability in existing mechanisms.

Keywords: software architecture; architecture principles; software sustainability; case study

Citation: Funke, M.; Lago, P.; Verdecchia, R.; Donker, R. . *Software* **2024**, *1*, 1–41. https://doi.org/

Received:
Revised:
Accepted:
Published:

1. Introduction

Increasing global concerns about climate change have raised interest in environmental sustainability in various research disciplines and industry sectors. The aviation sector, for instance, has been actively shifting to greener practices. A 2013 report found that the majority of European aviation players anticipate the impact of climate change on their operations by 2050 [1]. As a consequence, the Royal Schiphol Group[1,2], which is responsible for managing Amsterdam's Schiphol Airport, announced their vision for 2050 as "create the world's most sustainable airports" [2].

In approaching the concept of sustainability, we recognize the United Nations' fundamental definition as the "environment's ability to meet present and future needs" [3]. This global perspective has led to the emergence of the Sustainability Development Goals (SDGs)[3] as worldwide targets, along with actionable indicators to measure progress on a global scale. However, for industries that focus on specific impacts and outcomes, a more targeted approach is needed. In this regard, Wynn and Jones [4] uncover "an issue about how companies report their progress in addressing the SDGs", noting that "there is no generally agreed framework for companies to report on the SDGs". Reporting becomes particularly important in light of the enforced Corporate Sustainability Reporting Directive

[1] Royal Schiphol Group - https://www.schiphol.nl/en/schiphol-group/. accessed: 2024-01-30.
[2] In our study referred to as the *Schiphol Group*.
[3] SDGs - https://sdgs.un.org/goals. accessed: 2024-03-01.

(CSRD) by the EU in 2023 [5]. The CSRD sets high reporting standards for sustainability information for a wide range of companies. These requirements, together with the strategic sustainability goals defined by companies, identify the need to strengthen and monitor sustainability in an industrial context [6]. Key Performance Indicators (KPIs) have proven to be a useful tool for continuously monitoring project performance [7,8] and can provide the necessary feedback on whether strategic sustainability goals are being achieved [6, ch.4]. We therefore use KPIs to facilitate a systematic approach measuring sustainability in the context of software architecture.

Software architecture is defined as a major part of the software engineering discipline. It is described as a process organizing components to form an overarching system [9] and is the foundation "to reason about the system" [10, p.4]. In practice, software architecture is a process that affects an enterprise holistically, and thus emerges on different business layers as: (i) enterprise, (ii) application and data, and (iii) technology architectures [11]. A strategic transformation, like the vision of the Schiphol Group for 2050, requires instruments on all these layers to steer decisions towards the future. Within software architecture, principles are an instrument to fill in the gap between the different layers, *i.e.,* between the high-level strategic objectives and the specific design and implementation decisions [12]. Principles provide underlying rules and guidelines for realizing the strategic objectives [11,12]. In this research, we use the concept of architecture principles to effectively embed sustainability into the software engineering process, helping organizations achieve their strategic goals for a more sustainable environment. Thanks to the close relation of principles with the business layers, KPIs provide an effective mechanism to quantify the principles' impact on sustainability. Long-term observations in form of KPIs can help identifying the positive or negative effects and can therefore support the software architecture process.

To achieve sustainable development in the first place, four sustainability dimensions should be considered [13]: technical, economic, environmental, and social. However, current research and industrial practice discuss software architecture and its impact mostly on certain aspects of sustainability, such as environmental concerns only (*e.g.,* [14–16]).

In response to the problem statements above, the main objective of this research is to create a process for evaluating and monitoring the impact of architecture principles on the four sustainability dimensions. In addition, we aim to illustrate the application and integration of such process in professional practice. In cooperation with the Schiphol Group we address this objective by executing a case study and examining potential KPIs and their impact on sustainability.

In a previous study conducted with the Schiphol Group, Gupta et al. [17] proposed the Principle, Rationale, Strategies, Measures (PRSM) model to map a software architecture principle incorporating the four sustainability dimensions. In that work, the authors focused on the theoretical background and evaluated it based on various software architecture principles. However, the framework was developed without relation to an actual software solution and without a validation of the measures. In our present follow up study, we reuse and extend this PRSM model and evaluate it on a real-world software solution. The Schiphol Group is a suitable example of a large airport management company whose business strategy is already positioned towards sustainable IT and which is already using the sustainability framework of Gupta et al. [17] in practice.

Our main contributions consist of:

(i) a process pipeline to perform the sustainability analysis, *i.e.,* to obtain PRSM models and their extension. The pipeline associates architecture principles with sustainability quality attributes, KPIs, and measurement tools to monitor them;
(ii) the visualization of the derived measurements in form of two types of spider charts. The graphical representations can be used on the strategic, operational, and tactical level to derive a principle's performance on sustainability;
(iii) the application of the process pipeline and visualizations in a real-world context to draw conclusions for future studies.

This paper is organized as follows: Section 2 provides information and the necessary context to help understand the background. Section 3 presents our research questions and describes the research steps as well as the method we use to answer them. Section 4 outlines the industrial case study in detail. Section 5 documents the results of our study in form of the final sustainability models, KPIs, and measurement tools. An evaluation of the results is given by Section 6 in form of concrete measurements and the execution of a focus group. Section 7 presents an analysis and discussion of the results and reports potential threats to validity. Section 8 poses related work, and finally, in Section 9 we close the paper by summarizing the results and outlining possible future research.

2. Background

Table 1. Summary of the most important concepts relevant for the study at hand.

Concept	Full Name	Rationale	Reference
KPI	Key Performance Indicator	Define measurable business objectives and monitor the impact of architecture principles on sustainability.	Parmenter [8]
SMART	Specific; Measurable; Achievable; Relevant; Time phased	Evaluation criteria of the conditions and relevance of a certain KPI.	Doran [18]
SQ model	Sustainability-Quality (SQ) Model	Captures for each sustainability-quality concern definitions regarding their context.	Condori-Fernandez et al. [19]
D-matrix	Dependency Matrix	Identifying and uncovering missing sustainability-quality concerns.	Condori-Fernandez and Lago [20]
DM	Decision Map	Diagram framing and illustrating the sustainability-relevant design- and quality concerns and their related dependencies.	Lago et al. [13]
SAF-Toolkit	Sustainability Assessment Framework (SAF) Toolkit	A set of instruments (D-Matrix, SQ model, DM) to design the network of sustainability-quality concerns at the software architecture level.	Lago and Condori-Fernandez [21]
PRSM	Principle; Rationale; Sustainability Quality Attribute; Metric	Performs the sustainability analysis for one concrete software architecture principle by mapping sustainability quality attributes and KPIs. (tool-agnostic model)	Gupta et al. [17]
PRSM+T	Principle; Rationale; Sustainability Quality Attribute; Metric; plus Tool	Extension of the PRSM model by attaching tools necessary to capture concrete measurements for one particular KPI. (tool-dependent model)	*this present research*

In this section, we present the concept of measuring business objectives, the background in software and sustainability, and the groundwork on which our work is based on. Since we rely on a number of concepts that are relevant to this study, we provide an overview of the most important concepts summarized in Table 1.

2.1. Key Performance Indicators

In large organizations, KPIs can act as a fundamental management tool identifying gaps between the current situation and the aspired business and IT strategy goals, locating issues, and closing gaps [8]. In our study, KPIs provide a suitable mechanism to define measurable objectives and monitor the impact of architecture principles on sustainability. To evaluate the conditions and relevance of a certain KPI we make use of the SMART (Specific, Measurable, Achievable, Relevant, Time phased) characteristics. The SMART conditions were first introduced by Doran [18] to define effective business objectives. Ishak et al. [22] document how the SMART method became a widely-used concept and evolved to a mainstream method. Beyond the SMART evaluation of KPIs, Parmenter further describes seven characteristics of business KPIs [8,23] such as *measurement timing* or *responsibilities*. To the best of our knowledge, there is no current research that applies the SMART method or the characteristics from Parmenter on KPIs concerning software sustainability. In our research, we will showcase that both concepts can be successfully used also in such a context.

The Schiphol Group has already implemented various KPIs to continuously monitor its business performance and steer its processes. As IT is an enabler, helping a business to reach their business goals, the Schiphol Group has defined an *IT & Data strategy 2021 - 2023* to support achieving their vision. In our case study, we make use of this strategy to map existing goals (*e.g., use re-usable standardized building blocks*) and existing KPIs (*e.g., up-time for key platforms*) on software architecture principles regarding their sustainability impact. Nevertheless, we can only consider the available KPIs as *preliminary*, since they were developed with a different objective: to fill in the technology component towards the overall vision for 2050. In our research, however, we aim for balanced sustainability which includes the consideration of multiple sustainability dimensions. Moreover, we want to use KPIs to show how the impact on sustainability of a software system can be monitored—rather than the impact of the entire Schiphol Group. Therefore, it may not be possible to reuse all existing KPIs in their current form or benchmark them against existing measurements.

2.2. Software and Sustainability

Sustainability has been identified as a crucial part of software [13,24,25]. Towards IT sustainability, four dimensions [13] or five dimensions [25] have been identified, respectively. As the *individual*, the fifth dimension represents the well-being of an individual, we embed this dimension within the *social* sustainability dimension, and thus follow the approach and definitions from Lago et al. [13], as described below:

- The **Technical** dimension includes aspects about the implementation of a system and concerns about the evolution, maintenance, and long-term use of systems regarding software aspects.
- The **Economic** dimension refers to business concerns as capital investment and profitability to ensure capital.
- The **Social** dimension focuses on the concepts of embedding software systems into communities (*i.e.,* humans, groups, or organizations) to improve maintainability, trust, and quality of the software users.
- The **Environmental** dimension goes beyond CO_2 emissions and covers the effects of our actions on the natural ecosystem and the preservation of such to ensure long-term human welfare. [13,25]

Condori-Fernandez and Lago [20] characterize traditional quality attributes (QAs) according to the ISO/IEC 25010 SQuaRE [26] standard, and identify their contribution to sustainability. The output is two-fold: (i) all ISO/IEC 25010 QAs are mapped on the four sustainability dimensions to create a **Sustainability–Quality (SQ) model**, and (ii) dependencies between the QAs and dimensions are uncovered and quantified by providing a set of **dependency matrixes (D-matrix)**. The SQ model offers the possibility to express QAs related to a particular software project and to define the individual characteristics and impact on the sustainability dimensions. By defining the D-matrix, a QA can either has a contribution in two different dimensions (inter-dependency), or it can relate to a different QA within the same dimension (intra-dependency). The follow-up research from Condori-Fernandez et al. [19] combines the contributions outlined above in the form of the Sustainability-Quality Assessment Framework (SAF) Toolkit [21]. The **SAF-Toolkit** also incorporates **Decision Maps (DMs)** [27] to provide software architects with the necessary tools to holistically support decision making from a software sustainability perspective. We make use of the SAF-Toolkit as part of our sustainability analysis and defining sustainability QAs for architecture principles in a standardized way.

2.3. PRSM Framework

Gupta et al. [17] proposed a framework to map software architecture principles on all four sustainability dimensions. The authors redefined a strategic planning process model to link architecture principles to their sustainability concerns: the OGSM model[4] (Objective, Goals, Strategies, and Measures) was transformed into the PRSM model (Principle, Rationale, Strategies, and Measures). The framework was developed to establish a balance for a sustainable business and its services [17]. The Schiphol Group served as an example to derive software architecture principles on enterprise, solution, and domain levels; but the principles were not applied as part of a specific software solution.

For our research, however, the architecture principles and their analysis cannot be reused, as they do not apply to our chosen case. Beyond this limitation, the work from Gupta et al. [17] did not consider the ISO/IEC 25010 standard as a guideline to define software quality attributes. In comparison, our work aims at elaboration on the PRSM model incorporating the ISO/IEC 25010 standard. This standard is widely-used in professional practice, including by the Schiphol Group. The relevance of our work is also underlined by the future work suggested by Gupta et al. [17], which proposed to monitor architecture principles and their KPIs to determine the impacts of design decisions made. The PRSM model is therefore reused and evaluated for the first time in the context of a real-world software solution.

3. Study Design

In this section we describe the method of our research and the details of the study design. First, the overarching research questions are outlined. Then, the design of our study is reported by discussing all three research phases. To address the overarching research objective, we have derived a main research question (RQ). This RQ is further divided into two sub-questions, RQ_1 and RQ_2, as further documented below. As this research is conducted as an industrial study, we define the research questions within the context of a given organisation embedded in the aviation sector.

RQ *How can Key Performance Indicators of software architecture principles be operationalized and measured concerning sustainability?*
By answering this main RQ, we identify and evaluate options for measuring KPIs continuously in an industrial context. This will enable analysing and monitoring the impact of software architecture principles on the four sustainability dimensions over time.

RQ_1 *What tools are accessible to measure sustainability Key Performance Indicators for software solutions within a given organisation?*
Our goal is to identify a set of tools within the portfolio of a particular organization to measure KPIs in different sustainability dimensions. Since the tool portfolio is available beyond a specific software solution, the tools can also be applied to other solutions. It is common practice to measure KPIs in the technical and economic dimensions such as the number of bugs, the code quality or the net revenue. The goal is to derive tools also for the environmental and social dimensions.

RQ_2 *To what extent can the sustainability Key Performance Indicators be monitored in an automatic way?*
We use the tools identified in RQ_1 to investigate whether the KPIs can be monitored in an automatic way. Automation would allow continuous monitoring as well as continuous evaluation of the impact over time.

[4] The OGSM model is used in the strategic planning process to develop and document goals, strategic rationales, and accompanying actions to achieve precise and measurable objectives [28].

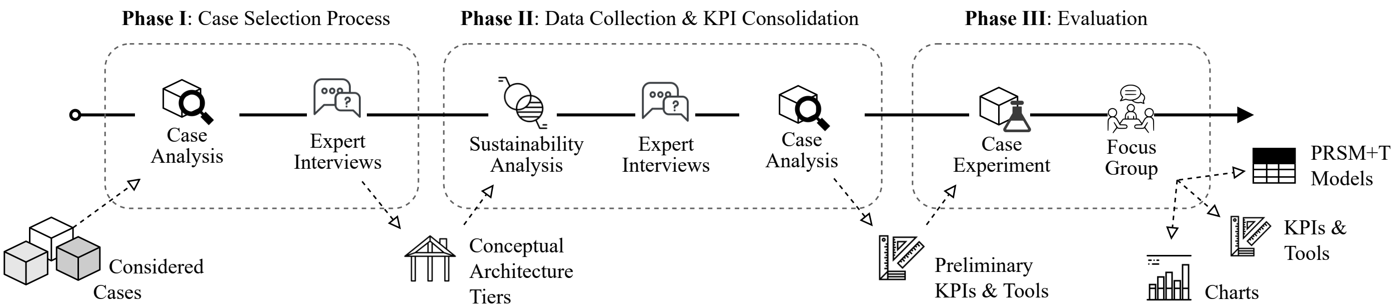


Figure 1. Overview of the Study Design.

Our research is organized in three phases. The overview of the study design is given in Figure 1 with explaining the individual steps on a high level in the following. A detailed examination of the steps involved are outlined in detail later on in Section 4.

Phase I

This phase is dedicated to the selection of the case under investigation. As stated in the objective of this research, we aim at defining a process to measure the sustainability impact of architecture principles. To that extend, we use a real-world software system to develop such a process. We consider only one software system as we want to derive and evaluate the process based on this case. According to Darke et al. [29], focusing on one specific case allows for an in-depth investigation and thorough comprehension of the desired methodology. Nevertheless, to generalize and strengthen our outcomes, additional cases need to validate and confirm the findings in contexts beyond the Schiphol Group [29,30]. The threats to validity in Section 7.2 discuss this in more detail.

To identify a suitable case for this research, we analyzed various software-intensive systems according to a set of systematic evaluation criteria which will be introduced later on. After the case is selected, the software system is studied in-depth. For that purpose, available documentations are scrutinized to gain familiarity with the software solution and to create an overview of its architecture. As main document we consider the architecture definition document (ADD). However, since not all information and background details can be part of such document, a second data source is used: expert interviews are conducted to validate and enrich the information extracted from the ADD.

As every single Schiphol Group software solution is driven by more than 20 architecture principles, we focus on the most influential ones to achieve targeted and analyzable results. Consequently, Phase I organizes the software solution according to its main architecture tiers based on the ADD and the internal organizational structure, *i.e.,* the project teams. Once derived, the tiers are revised by the experts during the interviews. Hence, the output of Phase I is one concrete case, structured by its conceptual tiers.

Phase II

We build on Phase I to determine the driving architecture principles associated with one particular tier. Additionally, we aim to distill associated KPIs (potentially) in all four sustainability dimensions particularly for the case under study. The proposed PRSM framework from Gupta et al. [17] is used in order to conduct the sustainability analysis and map the case-relevant architecture principles on all four sustainability dimensions. To underline the measurement tools required to monitor the associated KPIs, we introduce a dedicated column for the tools. We detach the tools (+*T*), as they are only an extension and are not necessarily needed for the analysis of the architecture principles themselves. While the PRSM model is sufficient to perform the sustainability analysis, the PRSM+T model focuses on an industrial context and is necessary to monitor the architecture principles over the long-term. In the remainder of our study we refer to the PRSM model as the *tool-agnostic model* and the PRSM+T model as the *tool-dependent model*.

The Schiphol IT & Data strategy 2021 - 2023 is consulted to identify preliminary KPIs. The potential set of KPIs, together with their sustainability mapping and related architecture principles, are used for a second round of expert interviews. Reviewing the interview results towards suitability for the selected case, a preliminary set of KPIs and measurement tools evolve. These KPIs and tools will serve as input for the final evaluation phase. The methodology of research Phase II, *i.e.,* the sustainability analysis, is concluded by providing a developed *Process Pipeline*. Such a pipeline is essential to define a standardized process for deriving a PRSM model for arbitrary architecture principles. We implemented this pipeline to complement the overall study design and the work of Gupta et al. [17]. It is also necessary in order to create comparable PRSM models across organizations in a systematic manner.

Phase III

Finally, we evaluate the obtained results by implementing the selected tools in the chosen case. Results are concrete measurements in the form of spider charts. This output is meant to help software architects and researchers monitoring sustainability KPIs. The measurements and visualizations serve as input for a final focus group to evaluate the results based on expert knowledge. The insights allow to present sound case study results along with reusable tools and KPIs. As a result, this phase provides (i) an extension of the PRSM model from Gupta et al. [17] to the PRSM+T model, (ii) a set of software sustainability KPIs and measurement tools, and (iii) a proposal to visualize the measurements in form of spider charts.

4. Study Execution

Our research follows the guidelines from Runeson and Höst [30] for conducting and reporting case study research in software engineering. Accordingly, a case study protocol together with a checklist is used to document each research phase and all case study design decisions [30,31]. Both are available in the online replication package[5].

4.1. Case and Subject Selection

Despite the observation from Runeson and Höst [30] that the case under study is usually intentionally selected, we opted for a systematic selection process to increase the replicability of our single-case research. Hence, all eligible cases from the Schiphol group are examined based on a list of criteria. Three criteria are derived from Runeson and Höst [30] (*i.e.,* C1 - Availability, C2 - Confidentiality, C3 - Case Size); the other three emerged from experience with industrial projects and are considered only for this research purpose (*i.e.,* C4 - Development status, C5 - Relevance, C6 - Completeness). The criteria and their description are outlined in the case study protocol as part of the replication package.

Initially, six cases are provided. As a detailed documentation of all the different cases is beyond the scope of this study and would not provide valuable insights to answer the research questions, only the evaluation of the actual case is presented in the online available case study protocol. After applying the criteria on the software solutions, we can conclude that all criteria positively contribute to the selection of the datahub platform *Port Community System (PCS)*. Only the large case size (C3) and the Proof of Concept (PoC) development status (C4) of the PCS solution can be partly considered as negative aspects. However, both criteria are considered to be a trade-off between (i) a wide range of available data and the extensive familiarisation period, and (ii) the limited feature set and the coherence with multiple architecture principles.

4.2. Case Description

The datahub platform PCS handles and integrates Cargo freight related messages from and to various parties for the aviation sector. Its main goal is to prepare, create and

[5] Replication Package - https://github.com/S2-group/MDPI_monitoring-sustainability_rep-pkg

keep track of the documents necessary for the transportation of goods from the shipper to the consignee. All involved customers and authorities can exchange data with each other and keep track of the status. The simplified architecture of the PCS solution is visualized in Figure 2. All provided information about the software solution itself, its architecture, and the functionalities are gathered by consulting the ADD and weekly tutorial sessions with software architects. Figure 2 highlights the interaction between the customers (*i.e.*, the freighters, ground handlers, and customs) with the airport.

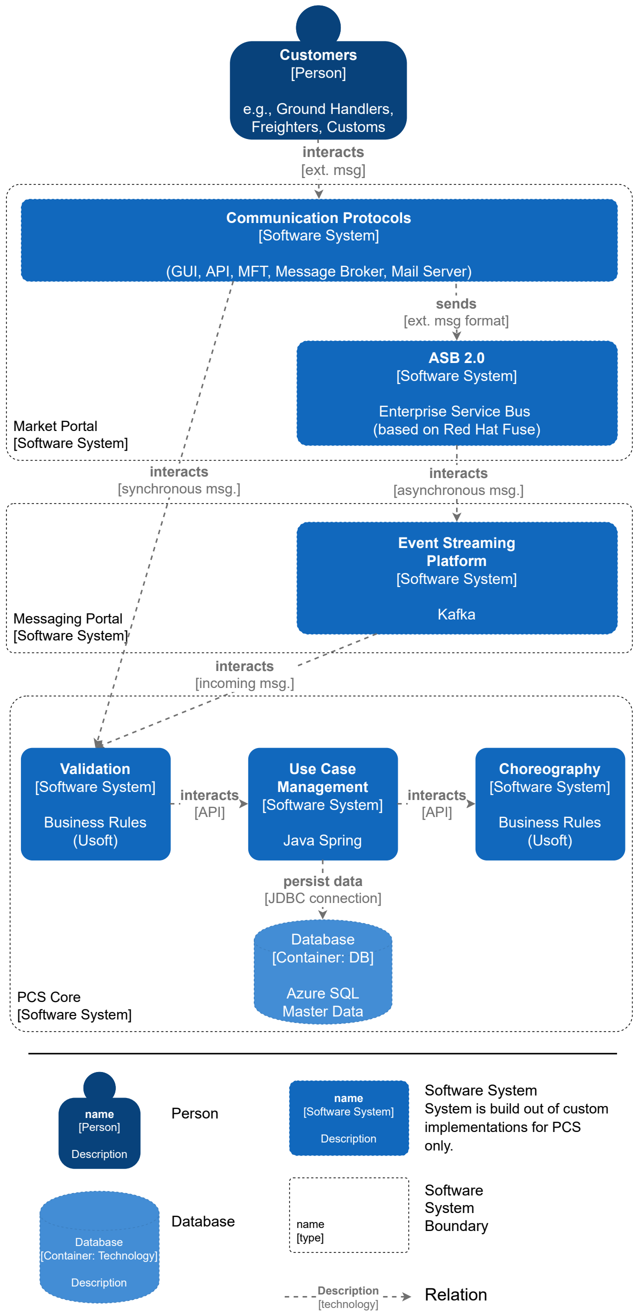


Figure 2. PCS Solution as high-level architecture view (C4-Diagram according to Brown [32]).

The description below outlines a general flow following the components depicted in Figure 2: the **customer** claims access to the system as it wants to create or request certain data into or from the PCS system. This access is done via various interfaces and communication protocols, *i.e., external data formats*. These protocols are (mostly) implemented as architecture building blocks. By relying on building blocks, the package of functionality can be ideally (re)used across software solutions and an organization [11, 33]. The external data formats need to be translated into an *internal data format*, valid specifically for the PCS solution. This translation is done at the **Market Portal** via the Airport Service Bus (ASB). The ASB is dedicated to implementing information exchange based on Enterprise Service Bus (ESB) technology. After the message has been translated, it is published as an event to the **Messaging Portal** where event consumers can subscribe to. Eventually, the message is processed by the **PCS Core**, responsible for outbound message orchestration, use case management, validation, and the persistent data storing.

4.3. Case and Units of Analysis

According to the definition of Runeson and Höst [30], we can consider the PCS solution as holistic case study with embedded units of analysis. In our research, the Schiphol Group serves as the context of the case study. Three different units of analysis are embedded and are examined, namely (*unit #1*) the tiers of the PCS solution, (*unit #2*) the driving software architecture principles, and (*unit #3*) the Schiphol IT & Data Strategy 2021 - 2023 with pre-defined strategic goals, metrics, and KPIs. By combining the case units of analysis, we can focus on the most driving parts of a software-intensive system (*unit #1*), perform the sustainability analysis on its architecture (*unit #2*), and map our findings to an actual business strategy (*unit #3*).

4.4. Expert Interviews

Two rounds of expert interviews are executed. Interviewees are invited according to their role and responsibility regarding the PCS solution (*cf.* Table 2). The initial contact with the experts was facilitated by the fourth author of our study, who possesses a network of contacts within the company and a comprehensive knowledge of each expert's role. This ensured a targeted recruitment process and enhancing the relevance of our experts to the PCS solution. In total, five participants are involved, divided in four interview sessions; P#2 and P#3 are interviewed at the same time as they are both key players concerning the PCS architecture. The interviews are conducted to increase precision of this research and are used as data triangulation to use sources beyond provided documents [30].

All interviews are designed as semi-structured interviews to provide as much flexibility as possible, but also to obtain replicable results. A mix of open and closed questions led to *funnel* interview sessions [30] by starting with open and broad questions and moving to more specific ones. Both interview rounds are described below; the full structure including all questions can be found in Appendix A.

Table 2. Interview partners corresponding their roles and responsibilities. **ID:** interviewee identifier; **Role:** current role of interviewee in the current company; **Responsibilities:** interviewee responsibilities regarding the PCS solution; **Experience:** interviewee industrial experience (in years)

ID	Role	Responsibilities	Experience
P#1	Software Architect	PCS Market Portal & PCS Messaging Portal	16
P#2	Enterprise Architect	General Architecture & PCS Core	32
P#3	Solution Architect	General Architecture & PCS Core	27
P#4	Cyber Security Officer	Governance & Security	21
P#5	Developer	PCS Core & Master Data	16

In **Round I** we identify the driving architecture principles for the case under research. Sub-objectives are to validate the previously defined tiers, gather first-hand knowledge about the PCS case and its stakeholders, and assemble potential QAs for sustainability. We aim to identify the driving software architecture principles concerning the selected tiers. To achieve this goal, the participants get asked the following main question:

> *What architecture principle(s) would you define as driving one(s) for this specific part of the PCS solution?*

In **Round II** we derive potential KPIs, gather universally valid measurement units, and explore a set of available or potential tools to measure the KPIs. The main question of this interview session is:

> *Regarding the PCS solution, what KPIs, metrics, and measurement tools would you define as applicable to this specific architecture principle?*

The results presented in this research, *i.e.,* the architecture principles, QAs, KPIs, and measurement tools are derived directly from the interviews. Since the experts are solely responsible for their allocated role and are solely interviewed about that role, the derived results can be directly attributed to the associated interviewee. Specifically, this means that the use of a particular coding strategy or a qualitative analysis of the interview sessions is not necessary. To address any potential gaps encountered during the interviews, available documents are consulted by the researcher (*i.e.,* the ADD and the Schiphol IT & Data Strategy) and then re-evaluated in the second interview session. Such interim steps are reported as intermediate results in the replication package which is publicly available.

4.5. Focus Group

As proposed by Kontio et al. [34], focus groups are a suitable tool related to the evaluation phase of a research; focus groups help answer questions like "what are the potential problems in using or understanding the model?". According to the authors, the focus group should be organized in three steps: (i) preparation, (ii) execution, and (iii) analysis. These steps are further described below.

Preparation

To follow the study design and comply with the typical size of focus groups (4-8 participants [34]), the same five experts interviewed in research Phase I and II are invited. This selection allows the experts to collectively evaluate the isolated results of the other participants from the previous interview phases. The focus group is structured in the form of presentation slides. The pre-defined questions are available in Appendix A. Main objective of this focus group is to:

> *Evaluate the final PRSM+T models with their measurements and their spider charts as tool to visualize sustainability.*

Execution

A "synchronous online focus group" [34] is conducted, which means that the participants are at different places at the same time and the group is computer-mediated by using Microsoft Teams as online-meeting tool. To provide a common setting, the session is opened by a short summary of the research topic. After this introduction, general rules (*e.g.,* time window, audio recording, etc.) are presented.

Analysis

After finishing the focus group, the recording is transcribed, analyzed, and reported. In contrast to the interviews, the aim of the focus group is to evaluate existing results. Hence, only the opinions and viewpoints of the experts on the final results are essential. We applied open and axial coding on the focus group transcript to achieve bottom-up coding and a synthesis of the observations (*i.e.,* inductive coding [35]). This means that no predefined coding categories are applied but the categories emerged from reading the transcript [35]. The procedure finally delivered four coding categories and five sub-categories as illustrated in Figure 3. According to these categories, the main observations of the focus group are discussed and evaluated in Section 6.3.

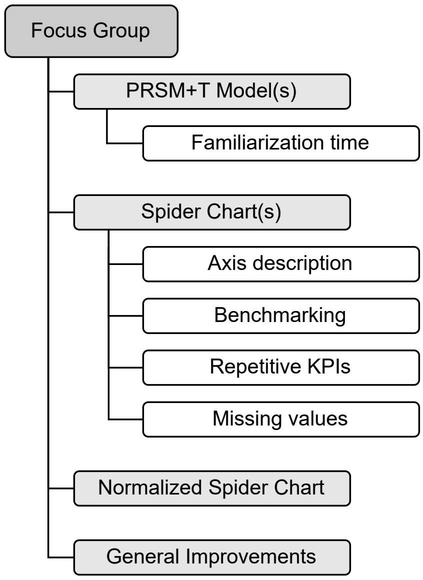


Figure 3. Focus Group coding categories.

5. Results

This section outlines the findings obtained with this research. All parts together are the result of applying the process pipeline. Specifically, four independent pipeline processes were performed: a separate process for each architecture principle. First, we introduce the selected PCS tiers together with their mapped principles. Second, the process pipeline is introduced considering three different levels of abstractions. At each level, the pipeline is examined at a different granularity to increase adoption beyond our specific case. Third, the developed PRSM+T model for one concrete architecture principle is explained in detail. Then, the used KPIs are discussed in detail. Last, all considered measurement tools are analyzed.

5.1. Architecture Principles

As described, the conceptual tiers are used to create a high-level abstraction of the PCS solution. By executing research *Phase I* and conducting the first interview session, the driving architecture principles according to these tiers were derived and are presented in Table 3. Throughout our research they will be used to (i) distil sustainability QAs, (ii) map KPIs, and (iii) depict suitable tools to measure the impact on sustainability.

Table 3. Final set of PCS conceptual architecture tiers, their description, mapped to the selected architecture principles and their rationale.

Tier **and Description**	***Architecture Principle*** **and Rationale**
PCS Market Portal Offers various options (*i.e.*, communication protocols) for customers to communicate with the PCS solution and send cargo related messages. The incoming external message format will be translated into an internal format.	*"Use the Airport Service Bus (ASB) for sharing / exchanging of operational data between applications and parties where routing, filtering, data transformation (integration rules) or transport transformation capabilities are needed."* The ASB is an integration platform as it adds functionalities to integrate two or more known systems. ASB incorporates routing, transformation, aggregation, throttling, basic reliable messaging and user management. However, the ASB causes more integration overhead due to increased data exchange as the number of connected applications increases.
PCS Messaging Portal Messages delivered via one of the communication protocols implemented at the PCS Market Portal are processed. An incoming message triggers the creation or an update of the cargo case.	*"SaaS goes above PaaS; PaaS goes above IaaS; IaaS goes above On-Premise."* SaaS solutions help reducing the cost and maintenance overhead of running cloud services. The technical knowledge does not need to be at company level and can be pass to the provider. This minimizes the risk of incidents. Nevertheless, it has to be ensured that the cloud solution comply with the company infrastructure and can be integrated.
PCS Core Responsible for the use-case-management, validation, orchestration, and persistent storing of cargo cases.	*"The system is made of loosely coupled components."* Many different communication protocols are supported to deliver or request Cargo related information. To be able to handle all kinds of communication, loosely coupled components are necessary. For instance, the PCS Core System is implemented in sub-components and is loosely coupled to the Business Engine which is implemented outside of the Core. Also, the responsibilities of the components are distributed across different layers; messages are used for communication between these layers (i.e., API).
Governance & Security Compliance with law regulations, Schiphol Group cyber security requirements, and Schiphol Group architecture principles to ensure security and safety across the PCS solution.	*"Always authenticate data flows and information requested by internal and external users."* The PCS implementation consists of multiple different components which need specific authentication and authorization capabilities. By following the "need-to-know" principle, user access controls and authorization procedures can be enforced. Its objective is to ensure that only authorised individuals gain access to information or systems necessary to undertake their duties.

5.2. Process Pipeline

To analyze the architecture principles and their impact on sustainability in a structured and reproducible way, a process pipeline was implemented and is presented in this section. In a first step, we describe the process on an abstract level. The abstraction reduces the pipeline to the underlying concepts without specifying concrete models. This allows adoption beyond the Schiphol Group, as all models can be replaced by other or similar ones—as long as the purpose is preserved. In the second step, we populate the actions and inputs with concrete models that will be used in our case study. In this step, we put the pipeline into a tangible environment and implement it into professional practice. In the final step, we conceptually tie the process pipeline into a general business context and illustrate how the pipeline and its output can be integrated into the decision making process.

5.2.1. Abstraction

Figure 4 shows the process pipeline on an abstraction level. All actions and inputs are described below.

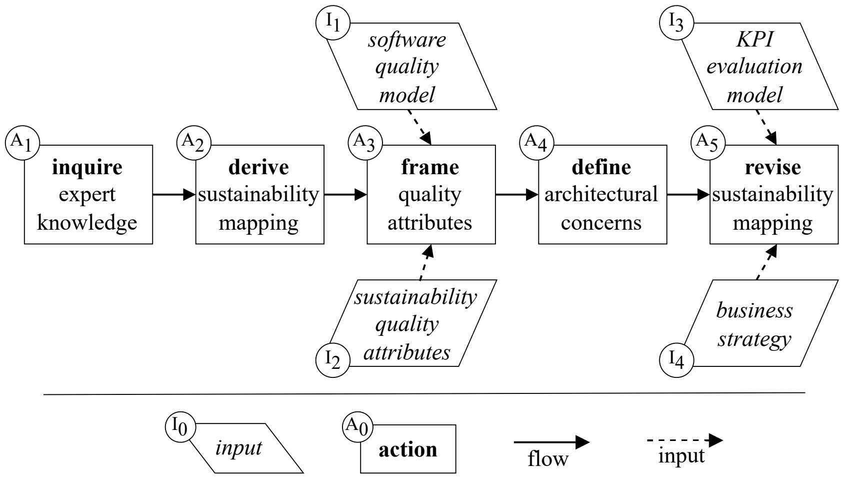


Figure 4. Process illustrating the abstract concepts containing **actions** ((A_1) - (A_5)) to perform the sustainability analysis and *inputs* ((I_1) - (I_4)) to support the actions.

(A_1) gathers the necessary knowledge about the software system under study. We use the knowledge of the experts, *i.e.,* the software architects, to determine which software architecture principle should be selected for the sustainability analysis and their rationale with respect to the software system. This information can only be derived from the experts involved in the development of the system, as only they are in a position to judge which principles are relevant. To our knowledge, there is no current model or architecture documentation strategy that documents the driving principles for a particular software system in a systematic way so that the information can be derived automatically.

(A_2) captures the expert knowledge in a preliminary sustainability analysis by mapping potential QAs to the selected principle and software system. We call this the preliminary sustainability analysis because the potential QAs should already be mapped to the four sustainability dimensions. Only when all four dimensions are considered, a balanced sustainability can be achieved. The analysis is considered as preliminary, since the QAs will be refined later in the process.

(A_3) models the preliminary QAs in a systematic way. For this purpose, we consider an arbitrary software quality model (I_1) together with a list of sustainability quality attributes (I_2). Both allow us to (i) uncover related QAs, (ii) identify sustainability-related quality concerns on all four dimensions, and (iii) uncover missing dependencies. Both inputs ensure replicability and comparability with other sustainability analyses performed with the same software quality model. The output of this step constitutes a model containing all related sustainability QA for one particular principle in form of, *e.g.,* a diagram.

(A_4) captures each defined sustainability QA definition regarding their context. Determining concrete definitions allows (i) the selected sustainability QA to be documented in a structured way for future assessment, and (ii) the selected sustainability QA to be reconsidered and revised.

(A_5) assigns KPIs and measurement tools to the sustainability QAs, resulting in a viable version of the sustainability analysis. The KPIs can either be derived from an existing business strategy (I_4) or developed from scratch. In either case, we suggest considering KPIs that contribute to a specific business objective - only then we can derive relevant information about whether the principle, and thus the software solution, is steering in the right strategic direction. We suggest applying a KPI assessment model (I_3) to evaluate the conditions and relevance of the selected KPIs.

All steps together will lead to a first working-version of the sustainability analysis. The analysis will focus on (i) the most relevant sustainability QAs, (ii) KPIs that measure the impact of the QAs, (iii) associated business objectives, and (iv) tools available to monitor the defined KPIs. The proposed pipeline can be repeated arbitrarily such that each repetition results in revised components (*e.g.*, revised concerns).

5.2.2. Implementation

Figure 5 illustrates the same process pipeline as described before, though showing concrete concepts used for implementing and applying the process in a real-world scenario. The preliminary PRSM model in A_2 represents the tool agnostic model and performs the sustainability analysis according to Gupta et al. [17]; while A_3 - A_4 follow the general usage-guidelines of the SAF-Toolkit from Lago and Condori-Fernandez [21]. A_5 concludes the pipeline by proposing the tool dependent model, *i.e.,* the PRSM+T model, and set the focus to a business and industrial context. Whereas the actions A_2 - A_4 and inputs I_1 - I_3 are based on existing deliverables and widely used standards, they were all developed either in isolation or without a software sustainability context. By combining and reusing these existing concepts, we are able to propose a reference process to obtain the sustainability analysis for architecture principles in a structured manner. All concepts are described below.

- A_1 For our case study, interviews with the five experts are used to derive the necessary knowledge.
- A_2 The interview results lead to a preliminary PRSM model. The model captures the architecture principle, its rationale, and assigns QAs while keeping the four sustainability dimensions in mind. At this stage, the model may include a set of multiple sustainability QAs for each dimension; ambiguities and uncertainties will be eliminated in the subsequent steps or another iteration.
- A_3 Decision Maps are used to model the driving sustainability QAs and uncover related quality concerns. As software quality model we consider the ISO/IEC 25010 SQuaRE [26] standard for defining QAs and framing the concerns in the decision map. To reveal dependencies between sustainability QAs and uncover missing sustainability concerns, the dependency matrix is used. As output we expect one coherent diagram, framing the related sustainability QA and revealing the driving ones.
- A_4 The concerns are captured in the SQ model to define their definition related to the case study context, *i.e.,* the PCS Cargonaut solution. As the SQ model is part of the SAF-Toolkit, it offers a central place together with the DM for documenting and preserving the sustainability analysis.
- A_5 All previous steps lead to a continuous revision of the PRSM model. Since all artifacts are relate to a corporate context, we consider the ADD and the Schiphol IT & Data Strategy to derive and map KPIs. To also capture the metrics and measurement tools necessary monitoring the KPIs, we use the tool dependent model, *i.e.,* the PRSM+T model to assign and highlight the measurement tools. The KPIs considered are analyzed according to the SMART method and revised to obtain sound KPIs.

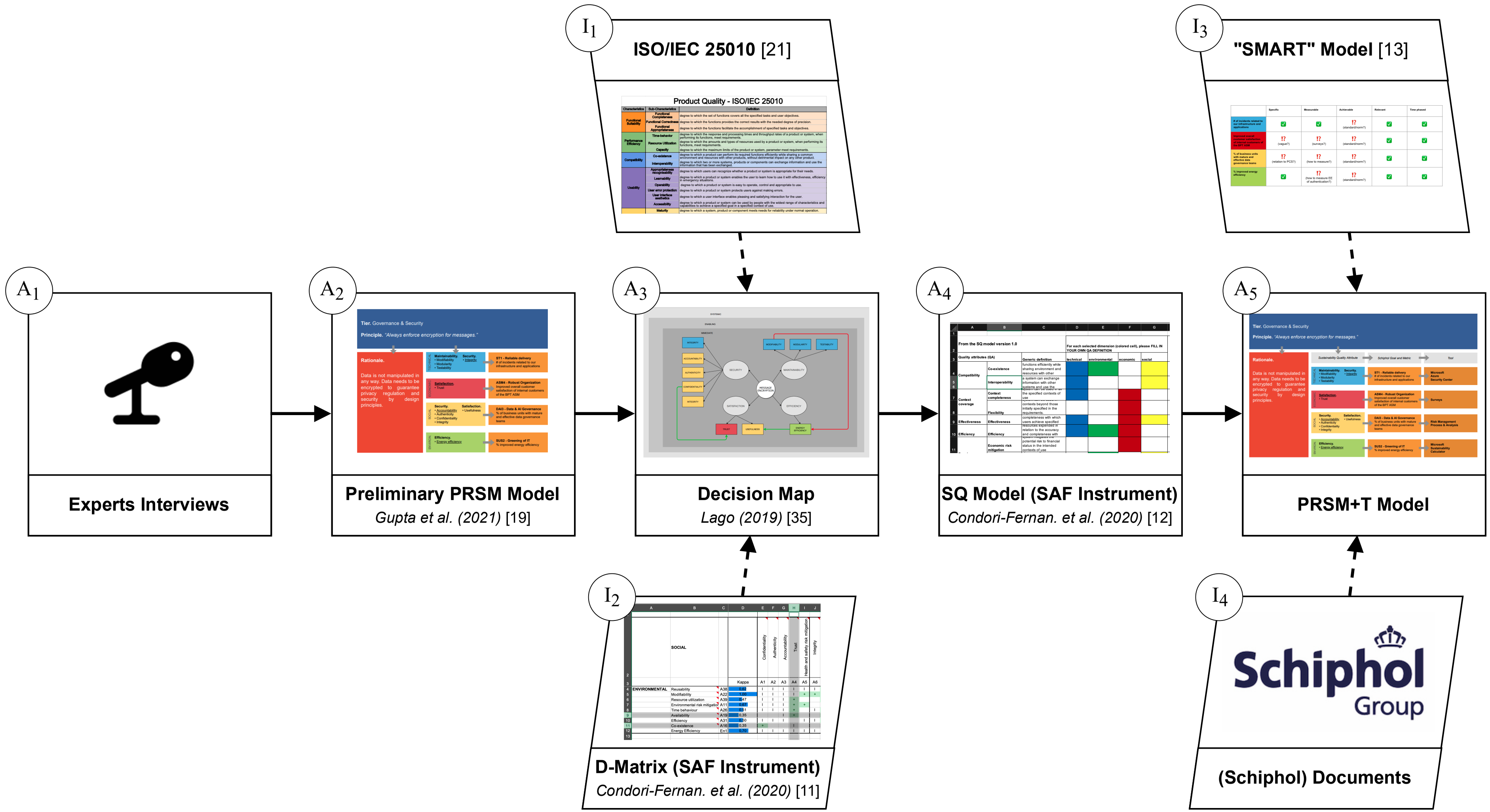


Figure 5. Process illustrating the applied tools and references used to create a PRSM+T model for a certain software architecture principle. **Actions** (A_1 - A_5) perform the sustainability analysis; *Inputs* (I_1 - I_4) supporting the actions.

5.2.3. Integration

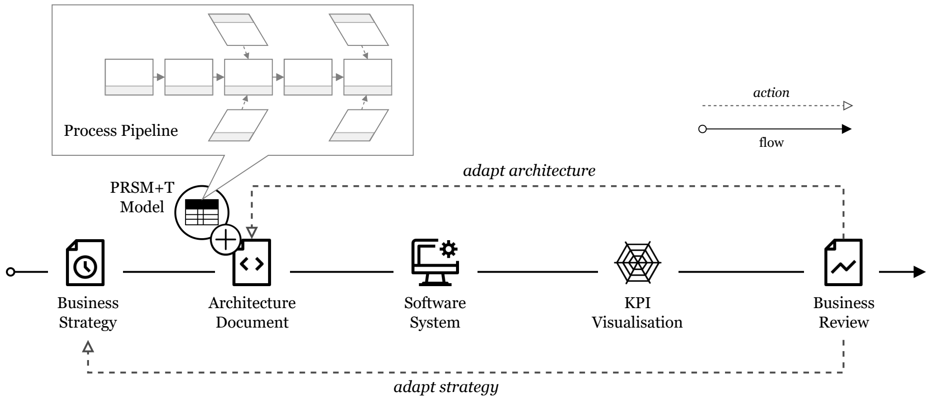


Figure 6. Integration of the sustainability analysis and process pipeline into a general business context to guide decision making.

Figure 6 illustrates the concept of incorporating the proposed sustainability analysis into the software architecture process with respect to the overall business strategy. As mentioned earlier, the principles of software architecture are derived from a specific business strategy and are used to guide the architecture process at all business levels. Using the proposed process, software architects can get guidance to create PRSM+T models and integrate them into the regular architecture process. Having a PRSM+T model as part of the regular architecture document enables the monitoring of the software system's impact on all four sustainability dimensions. Derived measurements in the form of visualisations can be fluently included into a regular business review. These reviews provide information on whether the software system implemented is steering in the right direction. If deviations are apparent that are not in line with the business strategy and the identified sustainability goals, actions ca be taken to adjust either the software architecture and its implementation (operational level)—or even the business strategy (strategic level), if necessary.

5.3. PRSM+T Model

As the aforementioned Table 3 shows, we covered the four main architecture principles related to the PCS solution in our case study. However, the following results are only shown in detail for the PCS Messaging Portal and its assigned architecture principle. The PCS Messaging Portal is selected as it contains the most relevant results, *i.e.*, interview observations and measurement data regarding the mapped architecture principle, for showcasing the entire workflow of the sustainability analysis. Focusing the presentation on a single tier allows us to provide an in-depth documentation and analysis of the results. Observations of similar nature can be drawn also for the other tiers. All information regarding the omitted tiers, *i.e.*, architecture principles, is provided in our replication package. The PRSM+T model (Figure 7), the DM (Figure 8), and the SQ model (Appendix C Table A5) are the final results of research Phase I and II including the interview sessions with P#1 and applying the pipeline.

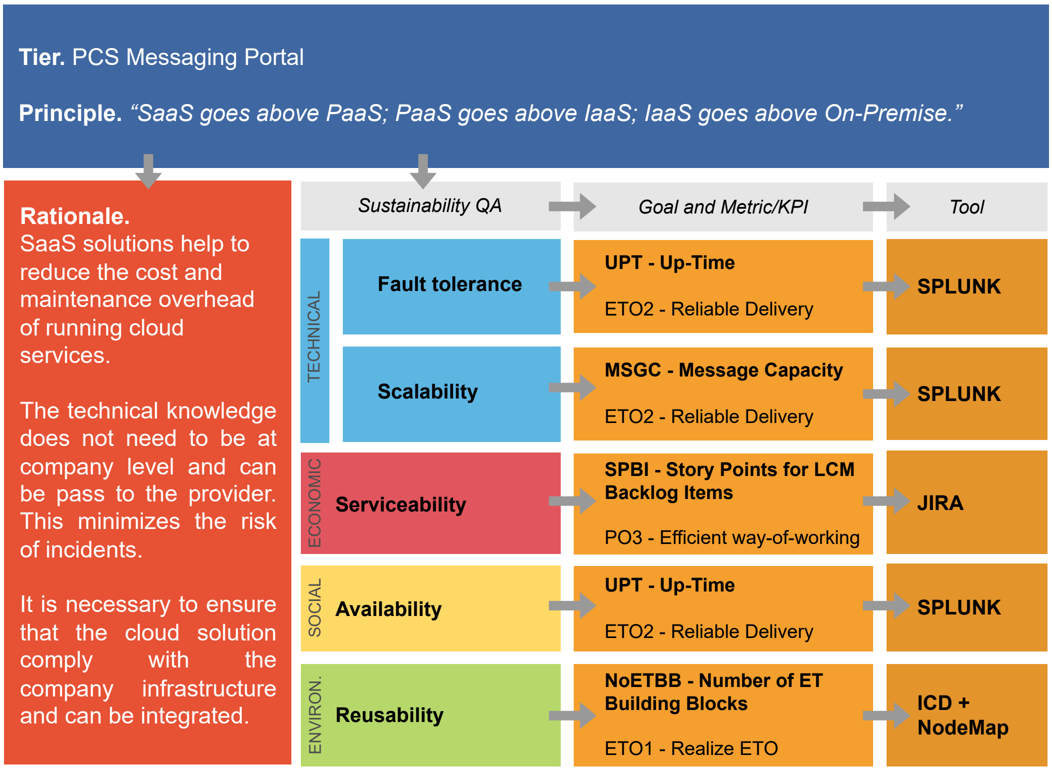


Figure 7. PRSM+T model - PCS Messaging Portal.

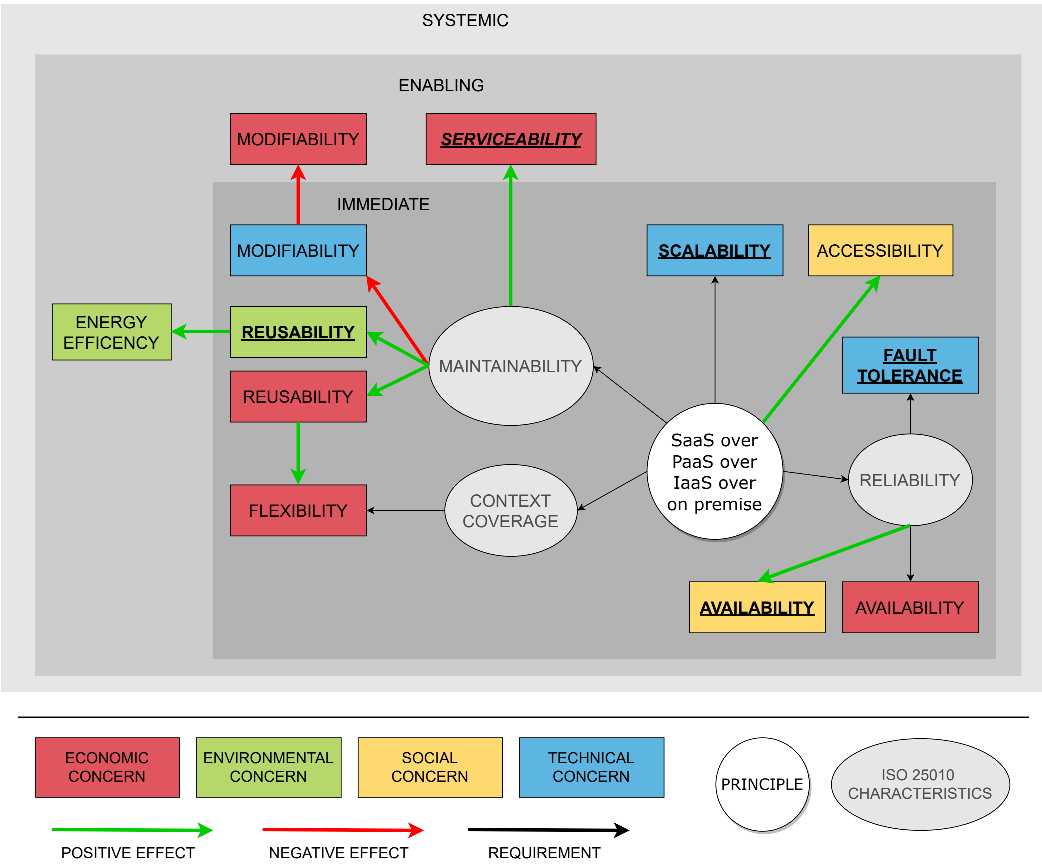


Figure 8. Decision Map for the PCS Messaging Portal illustrating the sustainability sub-characteristics and quality attributes. Underlined **concerns** are taken for the PRSM+T model.

The central part of the PCS Messaging Portal contains a message broker that is responsible for publishing and subscribing to streams of events. This event streaming platform is implemented by using distributed cloud solutions. Hence, it is not surprising that the cloud distribution principle was considered by interviewee P#1 as the driving one. While asking about the driving QAs for this architecture principle, P#1 explained that the architecture principle is also driven by the AMUSE characteristics:

- Adaptable: One size does not fit all.
- Maintained: Build once, run many times.
- Usable: Self Service, fits with needs.
- Sanctioned: Secured, tested and governed.
- Easy to start with: Get started in hours, not weeks.

These characteristics were used to further develop the actual QAs related to this architecture principle and software solution (*i.e.*, using the SAF-Toolkit and DMs). As can be observed in the PRSM+T model reported in Figure 7, a total of five QAs are distributed across the four dimensions that are discussed below:

The technical dimension (blue) contains two QAs: **Fault tolerance** and **Scalability**. This is an extension compared to the PRSM model from Gupta et al. [17]. Three out of five interviewees mentioned that sometimes it is not possible to distill the most important QA for a particular dimension. Hence, we allow to have multiple QAs for one single dimension $(1 - *)$. However, to preserve the focus on the driving QAs, we suggest to limit the amount of QAs per dimension to two $(1 - 2)$.

Serviceability (also referred to *supportability*) in the economic dimension (red) is considered as an outlier since it is not part of the ISO/IEC 25010 standard. Serviceability cares about maintaining the software system: *i.e.*, Life Cycle Management (LCM) like upgrades, updates, and the support beyond the development cycle. Hence, the QA is considered as sub-characteristic of the ISO/IEC 25010 characteristic *Maintainability*. In the context of SaaS solutions, serviceability is especially important for LCM as it is handled at the provider side. This ensures fewer support costs at the company side.

The social dimension (yellow) uncovers that only when the PCS solution is **available**, the customers will trust the software product and will use it eventually. In addition, in the DM in Figure 8 it can be observed that economic revenue can only be increased if the PCS solution is available. Due to this fact, the *Up-time* of the cloud solutions is considered as metric to measure both the fault tolerance in the technical dimension and availability in the social dimension.

Cloud solutions on the provider side can be shared among the customers. Thus, **Reusability** in the environmental dimension (green) enables reusable software solutions for multiple customers and saves resources; but does also save costs immediately when a cloud component can be reused across software solutions.

5.4. Key Performance Indicators

KPIs on their own do not give any information about a certain strategic goal. KPIs are only meaningful in combination with business goals and objectives [23]. Therefore, the KPIs necessary for the PRSM+T model were developed by consulting the Schiphol IT & Data Strategy 2021-2023. These KPIs and goals are depicted in the third column of the PRSM+T model in Figure 7. For example, the KPI *Up-Time* contributes to the goal *ETO2 - Reliable Delivery* which pursues continuity and an automated process between all involved parties in a reliable manner[6]. The SMART evaluation method was used to analyze all considered KPIs.

5.4.1. Final Set of KPIs

In total 14 KPIs were implemented across the four architecture principles and sustainability dimensions. As can be observed in Table 4, three different kinds of KPIs are available; the KPI was either: **(S)** extracted from the Schiphol IT & Data Strategy 2021-2023

[6] A detailed overview about the utilized Schiphol Group goals is given in Appendix B.

Table 4. Final set of implemented KPIs. **S**: reused Schiphol KPI; **S∗**: customized Schiphol KPI; ∗: KPI designed for Schiphol. **Goal:** assigned Schiphol goals (*cf.* Table A4). **QA:** mapped quality attribute(s) according to ISO/IEC 25010 **Dimension**: Technical ; Economic ; Social ; Environmental **Sorting**: grouped by dimension and then in an alphabetic order.

	KPI	Name	Definition (unit in bold)	Goal	QA & Dimension
∗	MSGC	Message Capacity	**Ratio (msg/min)** of total processed messages and a certain time period (i.e. one minute).	ETO2	Scalability (Technical)
∗	NoDaR	Number of Defects after Release	**Number** of defects of other systems than the changed system after a release was published.	ETO2	Modularity (Technical)
S	NoSI	Number of Security Incidents	**Number** of security incidents accruing on hosts which are involved into authentication.	ETO2	Integrity; Confidentiality (Technical)
∗	ToM	Throughput of ASB Messages	**Time** of message delay between in-going and out-going.	ETO2	Time behaviour (Technical)
S∗	UPT	Up-Time	**Ratio (%)** of total run-time and the total available time of the SaaS solutions.	ETO2	Faulttolerance (Technical); Availability (Social)
S∗	CpC	Costs per Change	**Costs (EUR)** per changes per component.	PR2	Effectiveness (Economic)
S∗	CRS	Cyber Risk Score	**Rating (0-100)** of the cyber risk based on the performed Business Impact Analysis (BIA) and the ratio of the normalized availability, integrity, confidentiality score.	CIO1	Economic Risk Mitigation (Economic)
∗	SPBI	Story Points for LCM Backlog Items	**Number** of story points for life cycle management (LCM) backlog items.	ETO2	Serviceability (Economic)
∗	NoOWN	Number of OpenShift Worker Nodes	**Number** of OpenShift Worker Nodes (i.e. Pods) grows with required system performance.	ETO2	Scalability (Economic); Scalability (Environmental)
S∗	NoCaDF	Number of Channels and Data Formats	**Number** of Channels and Data Formats available for data exchange.	ETO1	Interoperability (Social)
∗	NoTSV	Number of TLS/SSL Vulnerabilities	**Number** of vulnerabilities regarding TSL or SSL.	DAI3	Data Privacy (Social)
S	OCS	Overall Customer Satisfaction	**Rating (1-10)** of the overall customer satisfaction of internal and external customers as well as testers.	ASM4	Usefulness (Social)
S	NoETBB	Number of ET Building Blocks	**Number** of Enabling Technology (ET) building blocks that has been reused.	ETO1	Reusability (Environmental)
S∗	SRS	Security Risk Score	**Rating (1-5)** of the security risk based on the vulnerabilities regarding data relevant hosts.	CIO1	Health & Safety Risk Mitigation (Environmental)

and was ready to use; **(S*)** customized base on an existing Schiphol KPI, because it needed some optimization to fit our purposes, or **(*)** if no applicable Schiphol KPI was available, a dedicated KPI for this research purpose and Schiphol was designed.

When conducting the first round of interviews, preliminary KPIs were already derived. Interviewee P#4, for instance, stated that the Security & Governance department performs a survey of users within the organization to measure *awareness* once a year. This information was taken to find a correlated strategy goal as well as metric inside the Schiphol IT & Data Strategy. For this specific KPI, the metric *Overall Customer Satisfaction* could be found. Hence, the KPI was considered as *extracted unchanged* from the strategy to monitor *Usefulness* at the social dimension. The KPI *Cyber Risk Score (CRS)* illustrates an example where the predefined Schiphol KPI has to be adopted to fulfill the needs for the architecture principle. P#4 mentioned that the *Business Impact Analysis (BIA)* is an important tool to determine the CRS for a certain software solution. However, the strategy only defines the "cyber maturity based on the ISF Framework". In the second round of interviews this conflict was discussed and it could be concluded that the policies of the company are composed based on the ISF framework but the CRS is on a software solution level. As a result, the business goal and metric were taken but *customized* to measure the CRS. According to this procedure, three KPIs could be adopted unaltered, five KPIs were customized, and six KPIs were developed solely for this research.

As can be observed in Table 4, *ETO2 - Reliable Delivery* is the most frequently mapped goal; especially the technical dimension uses this goal exclusively. This can be attributed to the main purpose of the PCS solution and the selected architecture principles: as the PCS solution can be categorized as a datahub platform, its major objective is to receive, process, and deliver data. Hence, all related ETO building blocks need to be delivered reliable such that continuity and an automated process is ensured. This can be achieved by a transition to cloud applications. To monitor such a transition, KPIs are necessary (*e.g., UPT - Up-Time*; *NoOWN - Number of OpenShift Worker Nodes*).

5.4.2. SMART Evaluation

Table 5. KPI SMART Evaluation. ●: in full; ○: in part; ✖: not.

KPI	Name	S	M	A	R	T
MSGC	Message Capacity	●	●	●	●	●
NoDaR	Number of Defects after Release	○	✖	○	●	●
NoSI	Number of Security Incidents	●	●	○	●	●
ToM	Throughput of ASB Messages	●	●	●	●	●
UPT	Up-Time	●	●	●	●	●
CpC	Costs per Change	○	✖	○	●	●
CRS	Cyber Risk Score	●	●	○	●	●
SPBI	Story Points for LCM Backlog Items	●	✖	●	●	●
NoOWN	Number of OpenShift Worker Nodes	○	✖	○	○	●
NoCaDF	Number of Channels and Data Formats	●	○	●	●	●
NoTSV	Number of TLS/SSL Vulnerabilities	●	●	○	●	●
OCS	Overall Customer Satisfaction	●	●	○	●	●
NoETBB	Number of ET Building Blocks	●	✖	●	●	●
SRS	Security Risk Score	●	●	○	●	●

To evaluate the KPIs, the SMART assessment method is applied. Table 5 lists all KPIs and their evaluation. Each characteristic can either be (i) completely satisfied, (ii) partly satisfied, or (iii) currently not satisfied. For each SMART characteristic, we summarize our findings and observations below:

Specific

To some extent, certain KPIs are not that specific as initially thought or defined. This is most probably attributable to the fact that those KPIs are customized and designed specifically for the Schiphol Group. Hence, they do not currently have experience values from a longer productive operating phase and it cannot be concluded whether the KPI will be specific enough. A total of 14 KPIs are defined as fully specific, 3 KPIs only partially.

Measurable

As discussed in Section 4, the full feature set of the PCS solution was not available at the time this research was conducted. This also applies to some of the defined monitoring tools. Thus, all KPIs associated to a currently unavailable tool are defined as *currently not measurable*. Overall, tools were not available for 5 KPIs, 1 KPI could only be partially measured, and the remaining 8 KPIs supported full measurements.

Achievable

KPIs for which it is difficult to achieve the predefined standard are considered to be partly-achievable. This means that, for security-related KPIs, for example, considerable effort is required to achieve the norm. For the CRS, a norm of 0 was derived by the interview with P#4. However, a score of 0 is almost impossible to achieve, as every software solution involves some cyber risks and trade-offs. This is supported by the work of McKinsey [36], who states: "In most cases, it is impossible to stop all cyber attacks, so sometimes controls can be developed that tolerate some incidents". McKinsey recommends that business risks should be captured by defining dedicated Key Risk Indicators (KRIs) and linking them on KPIs which would lead to a "complete risk-based measurement". Due to this fact, 8 KPIs can only be partially achieved and 6 KPIs can be fully achieved.

Relevant

Only the KPI *NoOWN - Number of OpenShift Worker Nodes* is declared as partly-relevant towards providing more insight into the performance of the organization in obtaining its strategy. Due to the high degree of specialization and technology dependence (*i.e.,* OpenShift), the KPI addresses only a fraction of the entire IT landscape. The remaining 13 KPIs are considered fully relevant.

Time phased

All KPIs are completely time-phased. This is substantiated by the fact that the Schiphol IT & Data Strategy is time-phased in itself. For each year, quarter, and month, the company specifies and monitors the goals for every pillar by conducting reviews.

As explained in Section 2 and mentioned by Ishak et al. [22], certain KPIs do not necessarily satisfy all SMART conditions. This behaviour was especially observed for the *Measurable* condition, as not all KPIs are measurable to the point of this research. Only by having experience values from a longer productive operating phase, final conclusions can be derived.

If the characteristics from Parmenter [8] are considered, it can be concluded that the KPIs indeed violate some of these characteristics, because in detail, not all KPIs can be measured on a 24/7 basis. For instance, the *OCS - Overall Customer Satisfaction* cannot be monitored in such a way. Even if an automatic survey approach would be found, it is most likely that the satisfaction of customers does not change that frequently. While this study mainly considers non-financial KPIs, it also includes some financial KPIs (*cf. CpC - Costs per Change*), which violates the characteristic from Parmenter [8]. This can be explained by the fact that we aim to use balanced KPIs that can be used to monitor performance at all business levels and across all sustainability dimensions.

We can conclude that the characteristics from Parmenter do indeed help to revise and rethink sustainability KPIs in a software context. Using the example of the KPI *UPT - Up-Time*, the following revision was made: In the IT & Data strategy, the KPI *Up-Time* is defined as "Up-time for key platforms". However, by validating this KPI against the characteristic proposed by Parmenter, "the responsibility can be tied down to the individual or team", we can clearly deduce that "key platforms" constitutes an ambiguous definition

Table 6. Final set of implemented measurement tools. **Automation:** ✔: Completely; O: Semi; ✖: Currently not. **QA:** measured quality attribute(s). **Dimension**: Technical ; Economic ; Social ; Environmental **Sorting**: grouped by dimension.

Tool	Capability	Automation	QA & Dimension	
Splunk [37]	As "data-to-everything platform", Splunk offers various capabilities for logging, monitoring, and reporting for all different kind of data created on an application, server, and network level. We consider Splunk as key instrument to measure KPIs as it offers the most variety of possible measurements.	✔	[Technical] • Modularity • Time behaviour • Fault-tolerance • Scalability	[Social] • Interoperability • Availability
IBM Control Desk [38]	Provides monitoring for all information system layers. Hence, calculation of the *Number of Applicable Building Blocks* per software solution and *Security Incidents* can be retrieved.	✔	[Technical] • Integrity • Confidentiality	[Environmental] • Reusability
Business Impact Analysis	The BIA is used to systematically determine potential cyber security risks of a certain information system before implementing it (planning stage). The outcome is a Cyber Risk Score between 0 (best) and 100 (worst). Through external tools (*e.g.*, OneTrust, LLC.) automation is possible.	O	[Economic] • Economic Risk Mitigation	
Jira Software [39]	Jira itself does not consider actual financial values (e.g. € or $); instead, all values are implicitly related to financial values and indicated as *Story Points*. A story point refers to a certain number of labor hours and these, in turn, refer to an actual financial value.	✔	[Economic] • Effectiveness • Serviceability	
OpenShift [40]	Red Hat OpenShift offers a containerization platform for cloud computing. To monitor scalability in terms of worker nodes (*i.e.*, *Number of Pods*), the Monitoring API (*i.e.*, Prometheus) is used.	✔	[Economic] • Scalability	[Environmental] • Scalability
Qualys Inc. [41]	The tool enables auditing, cloud security, and compliance checking for IT infrastructures. We use the *Security Risk Score* computed for the hosts responsible for authentication and private data.	✔	[Social] • Data Privacy	[Environmental] • Health & Safety Risk Mitigation
Surveys	Used to systematically obtain information about attitudes, opinions, and behaviors of the people. They can be oral or written, structured or with open questions. Even though automatic survey tools are available, such surveys have to be created and interpreted manually.	✖	[Social] • Usefulness	

and involves at least more than one individual or team. Thus, to be more concrete and tie the KPI to a specific team, we have revised the KPI as "Ratio (%) of total run-time and the total available time of the SaaS solutions". However, inevitable violations of certain characteristics led us to conclude that the same observation from Ishak et al. [22] about the SMART method also applies to the attributes from Parmenter [8]: the criteria should be considered as a guideline, but do not necessarily satisfy all conditions; especially in the context of software and sustainability, violations can not be excluded.

5.5. Measurement Tools

In this section, the tools to monitor the KPIs across all four sustainability dimensions are presented. The capabilities of each tool, as shown in Table 6, were derived either during the weekly tutorial sessions with the PCS software architect or during the interviews. As defined in RQ_2, particular attention was paid to a potential automation of the monitoring process. Hence, Table 6 also outlines the ability for automation. It can be seen that five out of seven tools support automation completely, one tool provides only partial automation, and one tool does not support automation at all. In addition, the considered ISO/IEC

25010 quality characteristics are mapped to provide an overview what tool can be used to measure what QA.

Table 6 additionally depicts the instruments' ability to measure within the different sustainability dimensions. As shown, four tools support inter-dimensional measurements, while three tools are designated to one dimension only. The assignment to a sustainability dimension depends on the QA measured. It should be noted that all tools were already present in the portfolio of our selected organization and were used to measure the KPIs for the software system under study. It is certainly possible that a tool can also be used (i) in other dimensions, (ii) for other KPIs, or (iii) for other software systems. Moreover, all tools used in the Schiphol Group can also have a suitable equivalent in other organizations. Thus, we do not limit the set of potential measurement tools to the subset available at the Schiphol Group or to the chosen software solution. We rather provide our selection as a starting point for practitioners inside the aviation sector and beyond.

The large variety of tools could lead to increased complexity. This is also stated by interviewee P#4:

> *"It is really hard for us to have the right data at the moment when we need them. Therefore, we are looking for one dedicated tool to have all the data at one central point."* P#4, Cyber Security Officer

This issue was also identified during our research. Monitoring the KPIs through all seven tools leads to a considerable maintenance and development overhead. As each tool is related to its own administrative unit, the data necessary for this study needed to be retrieved from seven different sources.

6. Evaluation

By using the selected tools, concrete measurements were obtained to monitor the selected KPIs. The measurements were visualized in the form of spider charts and were presented to the final focus group aiming at evaluating the results based on expert knowledge. This section first examines the measurements. Then the conclusions drawn by the focus group are presented.

6.1. Case Study Measurements

Spider charts have proven useful for data analysis in business processes and benchmarking business performance [42]. Therefore, for each architecture principle, one spider chart was created[7] following the recommendations by Andersen [42]. Figure 9 visualizes all obtained measurements. Each axis represents one of the defined KPIs. The mapping between the architecture principle, the related sustainability QAs, the KPIs, and the tools can be seen in the final PRSM+T model in Figure 7. Despite other suggestions (*e.g.*, [43]), the spider charts created for this research (i) do not consider a unified point scale on each axis (*e.g.*, five-point scale) but do follow the suggestion by Andersen [42] to have a separate unit of measurement for each variable; (ii) the axis scales do not share a common minimum, because the center cannot be defined as a common zero point, since each axis has a different scale.

As mentioned, not all KPIs could be measured due to the development status of the PCS solution. Hence, the affected KPIs (7 out of 14) were marked as *n/a* and the value was set to 0. For all other KPIs, the value was obtained by using the corresponding tool and represents the factual value at the moment the data were extracted. As can be seen, for two charts (Figure 9a and 9b) it was possible to obtain real data for three out of five KPIs, one chart (Figure 9c) shows data for one out of five KPIs, and one chart (Figure 9d) does not contain any actual data. However, even with the missing values it can be clearly observed that the graphical presentation offers the possibility to keep track of the KPI metrics (a further discussion follows in Section 7). Future data sets in form of new data points would

[7] A detailed description of how the spider charts were generated (*i.e.*, programming language and code) including the final raw values can be found in the replication package online.

lead to a new polygon and, therefore, performance can be effortlessly monitored and benchmarked against previous data sets (blue polygon).

6.2. Normalized Spider Charts

Since architecture principles are subject to an iterative development process and a change in the business strategy can require the replacement of certain architecture principles [12], it is beneficial to accomplish a benchmark. A comparison of architecture principles allows to (i) keep track of the sustainability impact before and after a change or replacement, and (ii) uncover potential weaknesses in certain sustainability dimensions of the new or old architecture principle, respectively. Nevertheless, a comparison by using the spider charts proposed before is not possible. Due to different KPIs on each axis and the different number of KPIs in different dimensions, it is impossible to include a data set of one spider chart in another one and benchmark architecture principles against each other. To address such issue, Min-Max normalization [44] can be used to bring all variables to the same standing, *i.e.*, a scale of $[0 - 1]$. Min-Max normalization uses linear transformation to fit data into a predefined frame while preserving the relationship to the original data [44]. First, the *min* and *max* values are empirically derived to set the boundary; then, normalization on an arbitrary data set within this boundary is applied to rescale the entire range. The data set can then be used to visualize multiple architecture principles combined in one spider chart.

Throughout our case study, we were able to determine a snapshot of measurement data that represents the current state of the PCS solution. However, it is not possible to apply Min-Max normalization to a singular data snapshot (*i.e.*, one single data row) due to missing *min* and *max* values. Thus, randomized test data were used. To simulate a realistic data set, we generated 50 randomized data rows for each variable. After applying Min-Max normalization to the data set, we used the same spider chart visualization method to plot the data.

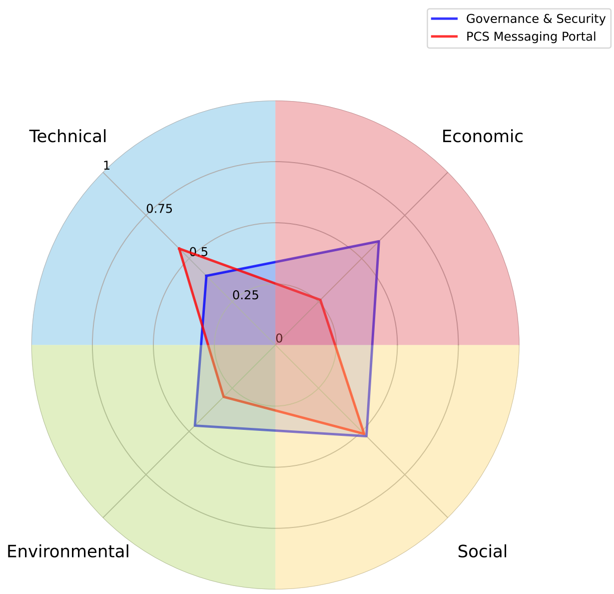
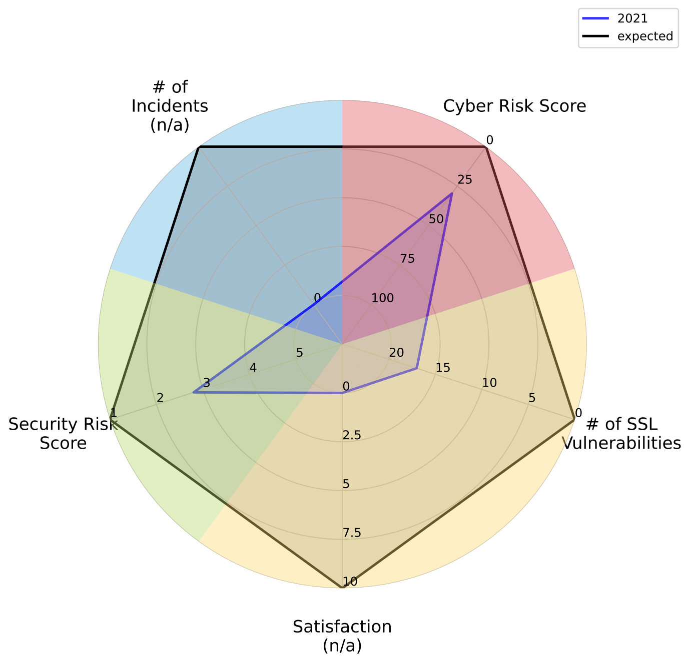
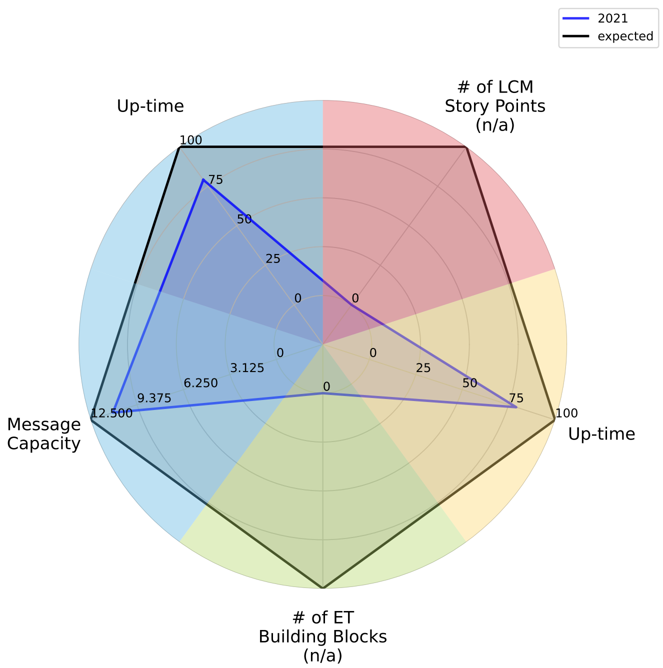


Figure 10. Example spider chart based on randomized and normalized data sets for both tiers: the *Governance & Security* (blue) and the *PCS Messaging Portal* (red).

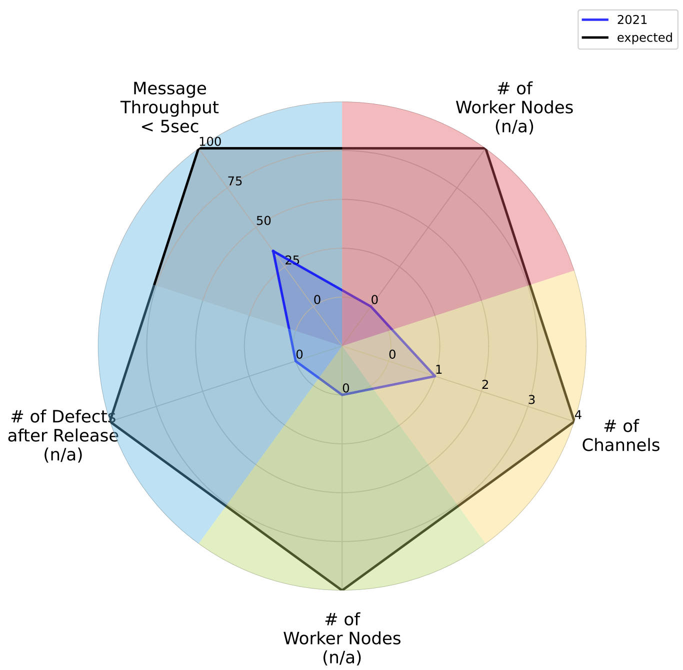
Figure 10 illustrates an example outcome of the previously described process, based on randomized data for the variables in *Governance & Security* and *PCS Messaging Portal*. We call this the *normalized spider chart*. Compared to the spider plots in Figure 9 (non-normalized spider charts) the normalized plot is now based on (i) a unified scale in the interval $[0, 1]$, (ii) a common minimum "0" in the center of the plot, and (iii) only one value per sustainability dimension.



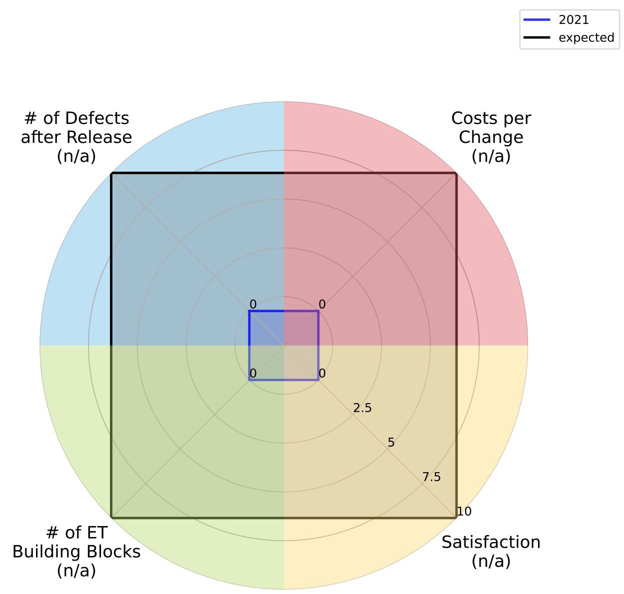
(a) Governance & Security



(b) PCS Messaging Portal



(c) PCS Market Portal



(d) PCS Core

Figure 9. Spider charts for all four PRSM+T models obtained from the PCS solution and the Proof of Concept (PoC) environment. **n/a:** Measurements for this KPI are not available and therefore set to 0. **expected:** The black outer polygon represents the expected values that can be achieved in the best case. **Dimension:** Technical ; Economic ; Social ; Environmental .

It can be concluded that normalization is necessary to visualize multiple architecture principles in one chart and to compare their impact in each dimension. However, if a detailed look at an architecture principle is necessary, the zoomed-in version (non-normalized spider chart) with all KPIs and their raw data would be necessary. As the normalization procedure does also come with disadvantages (*e.g.,* information loss), this kind of graphical representation was part of the focus group and will be discussed in the next section.

6.3. Focus Group Evaluation

Only four of the five focus group participants were able to attend the session - P#3 was hindered. For each architecture tier, the same group of questions were asked together with the derived case study results. For example, the PCS Messaging Portal was discussed together with its final PRSM+T model (Figure 7) and the resulting spider chart (Figure 9b). In the following, the main observations are outlined and discussed. We grouped the observations according to the uncovered coding categories.

Familiarization Time

Three out of four participants needed some time to reacquaint themselves with the presented PRSM+T models and spider charts, *e.g.,*:

> P#4 stated: *"I'm trying to understand the model. [...] You would need to explain it a little bit more to make it more understandable. So providing only the terms are a bit meaningless or hard to understand."* P#4, Cyber Security Officer

In contrast, P#1 did not need such familiarization. This could be because P#1 used such models frequently and was also involved in the development process of the PRSM model from Gupta et al. [17]. The latter can be considered a potential threat to validity and is therefore discussed in more detail in Section 7.2

Model Utility

All four experts considered the PRSM+T model in combination with a graphical representation as useful and beneficial for their daily business:

> P#4 stated: *"I do think that having such a model is quite helpful. [...] It will help us to understand what kind of things we are doing right or wrong."* P#4, Cyber Security Officer

> P#5 confirmed: *"I do also think that the analysis could help my department to keep track of their goals. Even if the model might need some learning."* P#5, Developer

Axis Description

Three out of four participants needed assistance regarding the terms used to describe the spider chart axis (*e.g., Message Capacity*).

> P#5 stated: *"It is not totally clear what you mean with Message Capacity in this context."* P#5, Developer

We observed that such naming might be difficult for non-experts to understand, since the terms are strictly related to the particular software solution; without a proper description, the meaning of certain axis and their values might be misleading.

Benchmarking

Intuitively, two experts compared the spider charts against each other regarding the performance of their KPIs (*e.g.,* by comparing the *PCS Messaging Portal* chart to the *Governance & Security* chart).

> P#5 described: *"this chart [the PCS Messaging Portal] performs better than the first one [Governance & Security]."* P#5, Developer

This statement shows that the graphical representations were used by the expert to *compare* two different models related to their sustainability impact. However, from a formal perspective, this intuitive behaviour is not correct as the different axis show different values and different value ranges (*cf.* previous discussion about normalization).

Repetitive KPIs

Using the same KPI in two different dimensions, *e.g., Up-time* in Figure 9b, (i) led to confusion among two respondents and (ii) could lead to a biased impact calculation as they show the same value but in different dimensions. From the focus group hence emerged that it might be necessary to avoid using two identical KPIs in the same model. This is contrary to what suggested in the literature, namely that KPIs should be reused as often as possible and as few KPIs as possible should be defined [23].

Missing Values

Missing values, *i.e., n/a* values, caused by the absence of data led to misleading interpretations, so that the performance of the overall architecture principle was interpreted as "poor" instead of "missing".

> P#2 stated: *"If we look at the throughput, it could give the impression that we still have much work to do."* P#2, Enterprise Architect

Nevertheless, this remark emphasizes how the spider charts are used by the experts: the current condition of a certain KPI (blue polygon) is compared to the desired value (black polygon).

Normalized Spider Chart

The focus group was also used to evaluate the additional *normalized* version of the spider plot (*cf.* Figure 10). This version of the spider chart was considered useful by all four attendees. Architecture principles compared to one another could be a useful tool.

> To this regard P#2 concluded: *"The normalized version could be the management summary, and the other ones are the detailed version to have a better and detailed look at it [...]. I think we could use both [...]. It shows you at which dimension we need to spend the money."* P#2, Enterprise Architect

> P#1 added: *"The management level would be also interested in the details, and would therefore need both versions of the charts because they want to know where exactly they need to put their money in."* P#1, Software Architect

Summary

From the focus group we can conclude that all experts found the graphical representation in the form of spider charts helpful. The experts used the charts to benchmark the architecture principles, intuitively. Moreover, the intuition of the experts led to the right conclusions, *e.g.,* that an architecture principle performs best when all KPIs match the outer polygon. These observations are consistent with the desired and also expected output of this research.

Improvements, however, can be made by: (i) changing the metric descriptions (*i.e.,* the KPI names) to a more common terminology; (ii) common upper and lower bounds for each metric would lead to a better understanding; (iii) KPIs that are used in two different dimensions in the same model could lead to confusions and should be reconsidered.

Applying normalization to the entire data-set result in a graphical representation that can be used to compare the impact of all architecture principles across all four sustainability dimensions simultaneously. The detailed, non-normalized version perform better at the operational level, revealing raw data and detailed information about which KPIs are falling behind; the normalized version have its strength at the strategic level as it provides a birds-eye view on multiple architecture principles and their impact on sustainability to find the right balance, even if some information is lost during the transformation.

7. Discussion

We present our main research contributions and the accompanying observations we made throughout this study by (i) interpreting the obtained results, (ii) comparing the results with the literature, and (iii) discussing the potential implications for researchers and practitioners. Possible threats to validity are outlined in the last section.

7.1. Contributions and Observations

The extension of the PRSM model [17] to the PRSM+T model helps measuring the impact of architecture principles on sustainability over long term. The model also served as groundwork to develop a process pipeline as outlined in Section 5.2. This pipeline defines the steps necessary to work with PRSM models practically and in a structured way. Thus, both researchers and practitioners are able to create replicable and especially traceable PRSM(+T) models.

Condori-Fernandez et al. [19] suggest to use the SQ model by defining plain definitions of the sustainability QAs under consideration. Despite this suggestion, the SQ models developed throughout this research (*cf.* Table A5) provide actual concerns over definitions. This can be explained by the execution of this research as a case study and the close relation to the industrial purpose. Throughout the weekly tutorial sessions, it was found that the defined sustainability QAs are always related to current business concerns. Therefore, the SQ models developed in this study can be successfully applied in practice as they reflect daily operations.

The PRSM model, its extension, and the process pipeline were applied and evaluated in a real-world scenario for the first time. Gupta et al. [17] evaluated the PRSM model based on five different architecture principles without relation to a specific software solution; in contrast, our research used and analyzed four concrete architecture principles related to the PCS solution. By conducting interviews and a focus group involving experts across different business units, the research results were evaluated. It became evident that the consideration of a software quality model (*e.g.*, ISO/IEC 25010 [26]) is of great importance to ensure compliance in industrial practice. Without following a standard, the comparison and re-use of the PRSM+T models is questionable.

PRSM+T Model & Process Pipeline

- We extended the static **tool-agnostic model** (PRSM) to a measurable **tool-dependent model** (PRSM+T).
- A **process pipeline** was implemented to systematically develop PRSM+T models.
- We **applied** both the PRSM(+T) model and the process pipeline in a **real-world context**.

As the usage of KPIs without considering a business strategy does not constitute meaningful information [23], the KPIs designed and used by this research are mapped on the IT & Data Strategy towards contributing to the overarching business goals. This mapping is also embedded into the PRSM+T model. The SMART analysis revealed that some KPIs (6 out of 14) are not yet measurable in our chosen case. This conclusion supports the assertion by Ishak et al. [22] that not all KPIs necessarily satisfy all SMART conditions. For example, in an early KPI development process the KPI might not be fully time-phased and the value might not be expressed in time until later. In addition, it may not be possible that one positive effect with a technology related target ensures also a positive effect in other measures as unknown technologies harbor always risks. To the best of our knowledge, the SMART analysis was used for the first time in the context of software-related KPIs to monitor software sustainability.

KPIs used by this research offer the capability for inter-dimensional support. This means that the same KPI can be used to measure the same (or even a different) QA in a different sustainability dimension. For example, in the context of this study this conclusion underlines the suggestion from Parmenter [23] to define as few KPIs as possible. However, the usage of the same KPI in the same PRSM+T should be re-evaluated. As mentioned by the focus group, using the same KPI twice in the same spider chart but in different dimensions can lead to misunderstandings and biases in benchmark calculations.

Overall, the SMART method and the characteristics by Parmenter [23] can be used as guidance to develop sound KPIs. The more precise the defined KPIs or targets are, the more focused the efforts can be and the greater the chances of achieving the goal [22]. We can conclude that KPIs are useful and necessary to monitor the impact of architecture principles on sustainability. Our proposed way of defining and assessing KPIs can be used in the future. In particular, practitioners can apply the process to develop own KPIs or even reuse some of our KPIs to keep track of their own architecture principles.

Key Performance Indicators

- We provided a set of **14 KPIs** including their contributions to Schiphol's IT & Data Strategy.
- **Mapping** of all KPIs on their related **sustainability dimensions** and **QAs** was applied.
- We observed that KPIs can monitor **inter-dimensional** performance (for different PRSM+T models).

We explored seven tools to monitor the KPIs in the real-world context of the PCS solution. All tools were already available at the Schiphol Group and could be reused. The KPIs and their measurement tools are suitable for use by the Schiphol Group as a method for measuring further architecture principles. The adoption of available tools and their support for automation implies that the sustainability analysis (i) can be easily applied and integrated into everyday operations, and (ii) is lightweight as it leverages existing capabilities within an organization.

Many different tools, however, lead to problems with consistency, as mentioned by the interviewees. To overcome the inconsistencies caused by different tools and the reliability, availability, and separation issues caused by centralizing data, as many KPIs as possible should be measured by available tools before introducing new ones.

Measurement Tools

- We provided a selection and analysis of **7 measurement tools** together with a mapping of their **inter-dimensional support**.
- We propose to **reuse** as many **centralized measurement tools** as possible to enable a lightweight sustainability analysis and prevent potential inconsistencies.

Spider charts were used for visualization as they provide an overview of performance levels for various indicators while revealing lagging variables [42]. In this study, color-coding was used to embed sustainability dimensions in the spider charts. All plots were created manually based on the data sets exported from the measurement tools. To support full automation, other tools are feasible (*e.g.,* Grafana[8]). Both, the experts from the focus group and the literature consider spider charts as a valuable tool to monitor business processes.

By applying Min-Max-normalization we created one common spider chart to compare multiple architecture principles simultaneously. As software solutions are implemented with consideration of all architecture principles involved, an performance aggregation of the principles would allow for comparisons among the various software solutions. In view of the mentioned issue of non-normalized spider charts, one could argue that a

[8] Grafana Labs - https://grafana.com. accessed: 2023-11-15.

trend analysis [42] might also be a suitable visualization to depict the performance of one architecture principle over time.

During the focus group session we derived the conclusion that spider charts are a useful graphical representation for keeping track of the sustainability impact. Nevertheless, the charts do also have downsides as the interviewees revealed: (i) continuous values without a maximum are difficult to interpret; and (ii) the mixed scales (*i.e.*, $[0-10]$ and $[5-1]$) may confuse non-experts. These two drawbacks support the use of the normalized visualization. However, the spider charts only depict the impact's general trend; the real business impact and risks remain hidden. One would need context-specific knowledge (*i.e.*, insights into the business in question) to translate the data into meaningful risks and their impacts.

Graphical Representation

- We used **spider charts** to monitor the impact of an architecture principle on sustainability and implemented the visualization of the corresponding **sustainability dimension**.
- A proposal is given of a **normalized spider chart** to compare the impact on sustainability of architecture principles against each other.
- We suggest to use the normalized spider charts on a **strategic level** as holistic overview, and to use the non-normalized spider charts on an **operational level** to zoom in and spot lagging KPIs.
- We observed that the actual **impact and risks** for the business are **not apparent**.

In this section, we have presented and discussed the results grouped by our main contributions. The results presented in this research are based on an industrial case study, and are therefore characterized by the attributes typical of such research method [30]. Therefore, our findings are positioned within a middle-ranged substantive theory [45], *i.e.*, the results gathered within the context considered in the study can be transferred to other contexts with similar characteristics. Throughout the design and execution of the research, the fit within a middle-ranged theory of both the presented process and the gathered results was purposely accounted for. First, we presented the PRSM+T model and process pipeline in their basic concepts so that practitioners and researchers can apply the model and pipeline to their own context by substituting elements as needed (*e.g.*, using a different software quality model). Second, the set of KPIs can be used by our case provider or by practitioners in other sectors as a starting point to integrate sustainability KPIs into their business strategy. Third, the measurement tools provided are generally accessible and thus context-independent, allowing them to be implemented by other software systems beyond company boundaries. Finally, the graphical representations in the form of spider charts can be used by both practitioners and researchers all domains. The visualizations are a generic mechanism to monitor and compare the sustainability impacts of principles. For the interested reader, further considerations regarding the generazability of the study are discussed in the following section.

7.2. Threats to Validity

This section analyses possible threats according to Wohlin et al. [46] (*i.e.*, threats to external validity, internal validity, construct validity, and conclusion validity). As this research was conducted as a case study, an additional threat to validity is considered as described by Runeson and Höst [30], *i.e.*, reliability.

External Validity

External validity reflects the validity of the results beyond our research and the relevance of the collected results to practice [46]. As hinted to in Section 7.1, given that the research we conducted is based on an industrial case study, it inherits the characteristics typical of such type of studies. Therefore, our results may be affected by generalizability threats discussed at large in the work by Runeson and Höst [30], *e.g.,* the population may not be representative due to the lack of statistics. For this reason, the results reported in this study have to be interpreted within a middle-range substantive theory [45], *i.e.,* the collected results only can be transferred to other contexts with similar characteristics. For this reason, we do not claim absolute generalizability of our results. In contrast, we consider the result collected in this study as the starting point, on which further studies considering similar or even different contexts can build upon to assess and strengthen the generalizability of the method. In other words, the work presents research-oriented results, on which further studies can build upon, by carefully considering and discussing related threats to external validity. To further mitigate potential threats and to ensure that the research results are relevant to practice, a state-of-the-art Schiphol Group software-intensive system was selected. A systematic evaluation was conducted to support the determination of the subject. The subject offers different architecture principles and a wide variety of available measurement tools; both help to mitigate bias, as we were not limited in our selection and analysis.

The *maturation effect* [46] of the research subjects, *i.e.,* the experts, can lead to bias if the interviewees are already familiar with the models or results being presented to them. To ensure that our research results were balanced on two levels of knowledge, two out of five experts were already familiar with the PRSM model and the topic of software sustainability; the remaining three experts were not familiar with either.

Conclusion Validity

Conclusion validity concerns the question of whether the conclusions derived were misinterpreted [46]. In our qualitative study, there could have been a risk that we or the respondents themselves could have drawn the wrong conclusions during the interviews. Potential issues may arise in the interview implementation as well as in their execution. To mitigate the *reliability of treatment delivery* [46] our interview sessions followed a predefined interview design that was cross-validated by the authors of this study. This ensured an identical interview process for all experts. However, since all experts belonged to the same organisation, we cannot rule out that respondents drew their conclusions in the best interest of the company and with less generalizable intent.

Internal Validity

Internal validity refers to the implicit assumption the independent variable is generally applicable and not driven by its context [46]. In our study, the process pipeline can be considered as the independent variable. Therefore, it should be emphasized that the process obtained is the result of conducting a single-case study and therefore cannot be declared as universally valid. The results are solely determined by the selected case, the corresponding principles and the associated experts. However, this single-case study was necessary to derive and propose this novel process pipeline in the first place. To mitigate risks related to this threat we rely on data triangulation and multiple data collection methods: we used evidence from (i) real-world documents like the ADD and the Schiphol IT & Data Strategy, (ii) related academic literature, (iii) quantitative data in form of real-world measurements, (iv) qualitative data from multiple expert interviews. The results were validated by conducting a focus group of experts with diverse professional backgrounds and an average of 22.4 years of industry experience. Nevertheless, to fully mitigate risks, the developed process should be applied in field studies.

Construct Validity

Construct validity concerns the extend to which the measures taken actually correspond to the intended concept [30]. Such concerns could arise during the interviews and are classified as *social threats* [46]. To mitigate these threats, first, the intermediate results were always presented to the experts and were part of the interviews to provide an additional validation of already obtained data; second, the final results were evaluated together by all interviewees during the focus group. The focus group ensures that the results obtained by *one* interviewee were also cross validated.

Reliability

We ensured reliability by designing a study geared towards providing replicable results [30]. Since our case study was conducted in the context of the aviation sector, not all raw data can be disclosed. In particular, safety and security-relevant data must be omitted. Nevertheless, we provide an online replication package including all the necessary resources to make our study transparent (*e.g.,* case study protocol, interview structures, intermediate results, and source code utilized).

8. Related Work

As in Section 2.3 described, our research builds on the work provided by Gupta et al. [17]. The main focus of Gupta et al. [17] is the sustainability analysis to map architecture principles on all four dimensions of sustainability by using the PRSM model. We extended the PRSM model to the PRSM+T model with related measurement tools. Additionally, we have analyzed a real-world software solution based on the PRSM(+T) model for the first time. To fill out such PRSM(+T) models in a common structure, we (i) first developed a process pipeline, (ii) then applied the pipeline in practice, and (iii) finally evaluated it with practitioners. Next to this fundamental groundwork from Gupta et al. [17], other research can be identified as related work and are discussed below.

Considering the evaluation of sustainability aspects in industry, a number of scholars have studied the role of sustainability in industry and attempted to integrate sustainability into a business strategy [47–50]. Chai [50] introduced the Sustainability Balanced Scorecard (SBSC) by extending the Balanced Scorecard (BSC) [48] through three sustainability pillars, *i.e.,* (i) economic, (ii) social, and (iii) environmental. Similar to the PRSM model [17] and our proposed PRMS+T model, the SBSC framework offers a multidimensional view of business performance by linking performance measures to goals. Hristov and Chirico [47] reused the SBSC model and proposed KPIs as suitable and quantifiable measures, to address and keep track of sustainability aspects. The authors also consider the selection process of appropriate sustainability metrics as one of the key problems in realizing sustainable systems. In contrast to the work of Chai [50] and Hristov and Chirico [47], in our research we also include the fourth dimension of technical sustainability and thus a relation to software concerns. Moreover, we overcome the problem of selecting appropriate sustainability metrics by providing a process to map sustainability KPIs to a real-world software solution using the PRSM+T model.

As IT and software are becoming ubiquitous in modern enterprises [12], the consideration of sustainability in software is gaining traction. Substantial research attention has been devoted seeking a definition of the term *sustainable software* itself and its meaning [24,51–54]. Early studies define sustainability either as the longevity of the software [15,55–57] or focus on environmental sustainability in terms of energy consumption [16,58–61]. A recent line of research has established that sustainable software can only be achieved holistically by addressing multiple dimensions of sustainability [13,20,25,62,63]. Venters et al. [25] emphasize the existence of dependencies and relationships between the different sustainability dimensions; potential trade-offs must be considered while developing the system. In our research, we are aware of such dependencies and consider these relationships in our proposed process pipeline, using the SAF-Toolkit and its dependency matrix [21]. Saputri and Lee [63] provide a comprehensive overview of the emerging definitions of software sustainability and complement the definitions with their limitations in terms of dimensions and potential metrics. The authors argue that most research only provide a "high-level abstraction" without concrete metrics and measurements. In contrast, our research provides metrics and measurements in the form of KPIs derived from a real-world software system and a process to systematically quantify sustainability. Moreover, we follow the holistic concept of sustainability by considering the four sustainability dimensions according to Lago et al. [13] and explicitly addressing possible interdependencies of these dimensions.

To incorporate sustainability into software several studies have been conducted on QAs and non-functional requirements [13,16,52,60,64]. Two different viewpoints can be derived from the recent body of research. While one view defines environmental sustainability as an additional non-functional requirement such as safety or security [64], the other identifies traditional quality requirements that contribute to sustainability and assigns these requirements to the sustainability dimensions [13,20]. To support wide industrial adoption, our research follows the approach from Condori-Fernandez and Lago [20] by relying on existing software quality models. Even though there is much work on addressing sustainability by software, there is limited research investigating actual measurement methodologies regarding software sustainability. While most of the work focus on the environmental dimension by quantifying the energy consumption of software [16,58,65,66], or focus on the technical dimension by using code metrics [14,55,67], less work have sought to capture sustainability on multiple dimensions [17,63]. The approach of Saputri and Lee [63] uses machine learning methods to assess sustainability criteria based on software code. Although the authors focus on three sustainability dimensions, *i.e.,* economic, social, and environmental, the analysis is limited to actual software implementation rather than software architecture. We aim to close the gap of sustainability QAs on software architecture by using a software quality model, *i.e.,* the SAF-Toolkit by Condori-Fernandez et al. [19], mapping KPIs on quality attributes and therefore consider all dimensions of sustainability regarding software architecture.

From the aforementioned number of studies, we can observe that an increasing attention is dedicated addressing sustainability aspects in software. However, in the current body of literature, only few studies investigate software sustainability from a software architecture viewpoint. Venters et al. [25] provide a comprehensive overview of available perspectives and terminologies on software architecture and sustainability, as well as a roadmap of recent research topics for sustainable software architecture. The authors, however, put the emphasis of their work on design decisions focusing on longevity. A number of other scholars have also focused on technical sustainability solely by discussing architecture longevity [15,55,56,68] and technical debt [56,69]. Ojameruaye et al. [57] proposed a method suitable evaluating technical and economic sustainability in software architectures. The authors seek to quantify sustainability dept of architecture design decisions. Nevertheless, the environmental and social dimension remain hidden. To support the design process towards holistic sustainability, Lago [27] provides decision maps to frame concerns considering all four sustainability dimension. In our study, we reused this concept of decision maps as part of our proposed process pipeline to map architecture principles on sustainability.

In this paper, we aim to overcome certain limitations of previous studies by: (i) taking a holistic view on sustainability; (ii) focusing on software architecture; (iii) quantifying sustainability QAs; and (iv) applying our research in an industrial context. Based on the groundwork of Gupta et al. [17] we aim at contributing towards a sustainable development in the context of software architecture by addressing sustainability holistically, *i.e.,* technical, economic, social, and environmental sustainability. Focusing on architecture principles allow architects to address and integrate sustainability on all different business layers. We use the notion of KPIs to quantify sustainability QA, opening up the feasibility of monitoring architecture principles over time. Our approach can be—and is already—embedded and applied in an industrial context supporting architects with necessary insights in their sustainability decisions.

9. Conclusion

To summarize our work and draw conclusions, we map our results onto the research questions as defined in Section 3. We close this paper by providing future directions for research.

RQ **- How can KPIs of software architecture principles be operationalized and measured concerning sustainability?**

To answer this research question, a single case study in the context of the Schiphol Group was conducted. Six different cases were considered and the datahub platform PCS was selected for this research. The general PRSM model can be used as a tool-agnostic model by researchers or on a strategic level to analyze architecture principles of sustainability. The extended PRSM+T model can be used as tool dependent model by practitioners or on an operational level to monitor KPIs with concrete tools. The proposed process can be used by both practitioners and researchers. Practitioners outside the Schiphol Group can apply the process by integrating the PRSM(+T) model into the architecting process and combining it with existing techniques. Even if an organization does not yet have elaborated KPIs, it can take our proposed KPIs as a starting point and use the process to develop own KPIs. Researchers can use the process as a reference to build upon or substitute certain steps with future work.

RQ_1 **- What tools are accessible to measure sustainability KPIs for software solutions within a given organisation?**

The KPIs and the tools were developed in a real-world environment. Therefore, the KPIs were measured with tools that are actually available at the Schiphol Group. As there is no universally valid tool that can monitor all KPIs by default, a set of tools was defined that can be used as a starting point by practitioners beyond case and organisational boundaries. In total, seven tools have been defined. The tools also support inter-dimensional measurements across the four sustainability dimensions. We can also conclude that existing tools in an organization should be reused to minimize the number of different data sources. Enterprise logging tools such as Splunk, for example, are useful for measuring multiple KPIs simultaneously. Therefore, centralized logging capabilities should be preferred.

RQ_2 **- To what extent can the sustainability KPIs be monitored in an automatic way?**

To answer this sub-question, all tools considered were analyzed according to their ability for automation. Most tools (six out of seven) support either full automation or semi-automation. Only surveys cannot be automated because of manual steps required. Nonetheless, surveys also have substantial value for monitoring sustainability as this research has shown: they are a key tool for the social dimension. Only by conducting surveys can the stakeholders' experience be measured.

Spider charts were used to monitor and visualize KPIs continuously. For each PRSM+T model, *i.e.,* architecture principle, one spider chart was created. Spider charts offer the ability to compare the impact of architecture principles over time and visualize all related KPIs in one plot. By applying normalization to the measurements, combined spider charts can be created that offer the ability to compare multiple architecture principles on all four sustainability dimensions for an entire software solution. While the normalized version can be used on the strategic level, the zoomed-in version offers a detailed view for the operational level.

9.1. Future Work

Our study was concerned facilitating an integrated monitoring process. A follow-up long-term study can use our solution to monitor and evaluate the KPIs over a long period. Such long-term study could be conducted on the PCS solution *in production*. The live environment would make it possible to implement all KPIs and tools as proposed deriving further insights. Derived data could help to explore specific relationships between particular architecture principles and certain sustainability dimensions by employing statistical significance tests.

Future research could also engage in presenting the proposed process to a wider and more diverse audience to assess its usability and generalizability. Potential improvements could be derived to integrate the pipeline even more in the daily architecting process.

A further extension of the present study could consider the implementation of the *ISO/IEC 2502n - Quality Measurement Division* [70]. This standard contains definitions and guidelines for elements of quality measurement. The close relationship to the ISO/IEC 2501n quality model used by our study would provide the opportunity to also define the measurement elements according to a well-known standard.

The list of measurement tools and the assignment on the sustainability dimensions could serve as the basis for a follow-up study. The purpose would be to derive general characteristics of tools for measuring software sustainability. The follow-up study could examine the characteristics and properties of state-of-the-art tools, classify them, and suggest ways to support sustainability.

As already envisioned in this study, spider charts can be used to derive further insights into the actual sustainability impact by calculating the area of the spider polygon. By examining the area it would be possible to draw further conclusions, such as the interdependencies between the sustainability dimensions and their KPIs. The answer to the question of the effective sustainability impact would still need to be investigated.

Supplementary Materials: The accompanying replication package can be downloaded at: https://github.com/S2-group/MDPI_monitoring-sustainability_rep-pkg.

Author Contributions: Conceptualization, Funke, M. and Lago, P.; methodology, Funke, M.; validation, Funke, M. and Donker, R.; formal analysis, Funke, M.; investigation, Funke, M.; resources, Lago, P. and Donker, R.; data curation, Funke, M.; writing—original draft preparation, Funke, M.; writing—review and editing, Funke, M. and Lago, P. and Verdecchia, R. and Donker, R. ; visualization, Funke, M. and Verdecchia, R.; supervision, Lago, P. and Verdecchia, R. and Donker, R.; project administration, Lago, P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Data are contained within the article and are available in the online replication package at: https://github.com/S2-group/MDPI_monitoring-sustainability_rep-pkg.

Acknowledgments: We would like to thank all interview participants from the Royal Schiphol Group who took part in this study and donated their time, knowledge, and support to make this research possible. We would also like to thank the Royal Schiphol Group for providing the case under research.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Interviews 1213

Table A1. Questions for interviews in research Phase I.

Goal: *Defining the driving architecture principle(s) for a certain part of the PCS solution.*

IQ1.1 What architecture tiers would you define to structure the PCS solution?

IQ1.2 Would you confirm that the following pillars represent the major components of the PCS solution sufficiently?

IQ1.3 Would you add/change certain pillars?

IQ1.4 What are the responsibilities and duties of the column for which you are in charge?

IQ1.5 Which main stakeholder(s) are involved/addressed by this tier?

IQ1.6 Which ETO building-blocks and/or cloud solutions (Saas/PaaS/etc.) are used for the implementation of this tier, and why? 1214

IQ1.7 What architecture principle(s) would you define as driving architecture principle(s) for this specific tier? And why?

IQ1.8 Which sustainability quality attribute(s) would you select as the driving attribute(s) for the prior selected architecture principle?

IQ1.9 For each selected sustainability quality attribute, can you define KPIs to track its impact in the context of the PCS solution?

IQ1.10 The KPI mentioned, how can they be measured (tools; automated; manual; surveys; etc..)?

Table A2. Questions for interviews in research Phase II.

Goal: *Defining applicable KPIs, metrics, and measurement tools to a certain architecture principle.*

IQ2.1 Do you agree with the selected architecture tiers defined during the first interview session?

IQ2.2 Do you agree with the selected architecture principles selected during the first interview session?

IQ2.3 Do you agree with the selected sustainability quality attributes selected during the first interview session?

IQ2.4 Do you agree with the revised and selected as driving sustainability quality attributes after using decision maps? Would you choose different?

IQ2.5 What KPIs and metrics can you think of to measure these particular quality attributes?

IQ2.6 Are you aware of the Schiphol IT & Data Strategy 2021-2023? 1215

IQ2.7 *(IQ2.6 answered with yes)* What Schiphol goal, metric, KPI is suitable to measure these particular quality attributes? Can you think of alternatives?

IQ2.8 *(IQ2.6 answered with no)* Do you think that the selected Schiphol goal; metric; KPI is suitable to measure these particular quality attributes? Can you think of alternatives?

IQ2.9 Which measurement tools would you suggest to use to measure these particular quality attributes?

IQ2.10 What do you think about this pre-selected measurement tools?

IQ2.11 Do you have final remarks about the just created PRSM+T model for the selected architecture principle?

Table A3. Questions for focus group in research Phase III.

Goal: *Evaluating the final PRSM+T models with their measurements and their spider charts as tool to visualize sustainability.*
For each group of PRSM+T and computed spider chart: IQ3.1 Is the final PRSM+T model well defined (mapping of the architecture principle to the sustainability quality attributes, the KPIs, metrics, and measurement tools)? IQ3.2 Can you confirm the relation to the spider chart? IQ3.3 What does the data (*i.e.*, spider chart) mean to you? IQ3.4 Do you see potential problems in using or understanding the model and the spider chart?
Regarding the normalized spider chart based on randomized data: IQ4.1 What are the advantages and potential problems in using or understanding the normalized spider chart?

Appendix B. Schiphol IT & Data Strategy 2021-2023 – Goals

Table A4. Utilized Schiphol IT & Data Strategy 2021 - 2023 goals for mapping the specific KPIs.

Focus on Business Value *ASM4 - Robust Organization.* Improve the focus of the customers to increase the business value.
Solid Foundation *DAI3 - Data & AI Governance.* Enable data governance to ensure appropriate data quality, lineage, as well as compliance with GDPR and ethical principles. *ETO1 - Realize enabling technology outlook.* Transform the basic technology landscape into standardized building blocks to assure reusable components. *ETO2 - Reliable Delivery.* Ensure a reliable transition to cloud applications to deliver continuity and automated processes.
Efficient IT & Data management *CIO1 - Safe & Secure.* Set cyber security standards to reduce the risk of cyber security threats. *PR2 - Increase efficiency.* Increase efficiency to reduce cost.

Appendix C. SQ Model

Table A5. SQ model - PCS Messaging Portal.

ISO/IEC 25010 Quality Model		Sustainability Dimension			
Characteristics	Attributes	Technical	Environmental	Economic	Social
Maintainability	Modifiability	SaaS solutions can not be easily modified due to the provider dependencies. Modifications might be not possible due to provider restrictions.		The SaaS provider takes care of the modification, hence custom modifications might be either not possible at all or expensive.	
	Reusability		SaaS solutions can be reused by more than one system, even over customers and can be part of other software solutions. Resources at the provider side can be shared.	If SaaS components can be reused across software solutions, costs can be reduced.	
	Serviceability			LCM for SaaS solutions is handled by the provider, hence, less support employees are necessary. System has attributes that make it easy to maintain beyond the software development cycle. It continues even when the software is no longer used.	
Context coverage	Flexibility			SaaS solutions can be used in contexts beyond the PCS solution. SaaS solutions have the ability to match with business needs as they flow [71].	
Reliability	Availability			System, i.e. the SaaS solution needs to be highly available. If not, delays in the Cargo process can occur, leading to flight delays and thus enormous economical costs.	If the system is not available, the users and customers do not trust and do not use the software solution.
	Fault tolerance	Even in case of software or hardware faults on the provider side, the SaaS solution would/should operate as usual due to redundancy on the provider side.			
Accessibility	Accessibility				SaaS solutions are usable by users with different disabilities [71]. This leads to access by many different user groups and with many different devices. In addition, the access to SaaS solutions are easier which decreases the barriers to the service.

References

1. European Union Aviation Safety Agency. *European aviation environmental report 2019*; Publications Office, 2019.
2. Royal Schiphol Group. Annual Report 2020. Available online: https://www.schiphol.nl/en/schiphol-group/page/archive-annual-reports/ (accessed on 2024-01-30).
3. on Environment, U.N.W.C.; Development. *Report of the World Commission on Environment and Development: Our Common Future*; 1987.
4. Wynn, M.; Jones, P. Industry Approaches to the Sustainable Development Goals. *International Journal of Environmental Studies* **2022**, *79*, 134–148.
5. European Parliament and the Council of the European Union. Directive (EU) 2022/2464 of the European Parliament and of the Council of 14 December 2022 Amending Regulation (EU) No 537/2014, Directive 2004/109/EC, Directive 2006/43/EC and Directive 2013/34/EU, as Regards Corporate Sustainability Reporting. Technical Report Directive (EU) 2022/2464, 2022.
6. Zipse, O.; Hornegger, J.; Becker, T.; Beckmann, M.; Bengsch, M.; Feige, I.; Schober, M., Eds. *Road to Net Zero: Strategic Pathways for Sustainability-Driven Business Transformation*; Springer International Publishing: Cham, 2023.
7. Sürücü, C.; Song, B.; Krüger, J.; Saake, G.; Leich, T. Establishing key performance indicators for measuring software-development processes at a large organization. In Proceedings of the Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. ACM, 2020, pp. 1331–1341.
8. Parmenter, D. The new thinking on key performance indicators. *Finance and Management Magazine* **2006**, pp. 1–4.
9. Kruchten, P.; Obbink, H.; Stafford, J. The Past, Present, and Future for Software Architecture. *IEEE Software* **2006**, *23*, 22–30.
10. Bass, L.; Clements, P.; Kazman, R. *Software architecture in practice*, 3rd ed.; SEI series in software engineering, Addison-Wesley, 2013.
11. The Open Group Architecture Forum. *The TOGAF® Standard, Version 9.2*; Vol. 9.2, 2018.
12. Greefhorst, D.; Proper, E. *Architecture Principles: The Cornerstones of Enterprise Architecture*; Springer, 2011.
13. Lago, P.; Koçak, S.A.; Crnkovic, I.; Penzenstadler, B. Framing sustainability as a property of software quality. *Communications of the ACM* **2015**, *58*, 70–78.
14. Koziolek, H. Sustainability evaluation of software architectures: a systematic review. In Proceedings of the Joint ACM SIGSOFT Conference on Quality of software architectures (QoSA) and Architecting Critical Systems (ISARCS). ACM Press, 2011.
15. Durdik, Z.; Klatt, B.; Koziolek, H.; Krogmann, K.; Stammel, J.; Weiss, R. Sustainability guidelines for long-living software systems. In Proceedings of the International Conference on Software Maintenance (ICSM). IEEE, 2012, pp. 517–526.
16. Kern, E.; Dick, M.; Naumann, S.; Guldner, A.; Johann, T. Green Software and Green Software Engineering – Definitions, Measurements, and Quality Aspects. In Proceedings of the International Conference of ICT for Sustainability (ICT4S), 2013.
17. Gupta, S.; Lago, P.; Donker, R. A Framework of Software Architecture Principles for Sustainability-driven Design and Measurement. In Proceedings of the International Conference on Software Architecture Companion (ICSA-C), Stuttgart, Germany, 2021.
18. Doran, G.T. There's a S.M.A.R.T. way to write management's goals and objectives. *Management review* **1981**, *70*, 35–36.
19. Condori-Fernandez, N.; Lago, P.; Luaces, M.R.; Places, Á.S. An Action Research for Improving the Sustainability Assessment Framework Instruments. *Sustainability: Science Practice and Policy* **2020**, *12*.
20. Condori-Fernandez, N.; Lago, P. Characterizing the contribution of quality requirements to software sustainability. *Journal of Systems and Software* **2018**, *137*, 289–305.
21. Lago, P.; Condori-Fernandez, N. The Sustainability Assessment Framework (SAF) Toolkit: Instruments to Help Sustainability-Driven Software Architecture Design Decision Making. Available online: https://github.com/S2-group/SAF-Toolkit (accessed on 2024-01-30).
22. Ishak, Z.; Fong, S.L.; Shin, S.C. SMART KPI Management System Framework. In Proceedings of the International Conference on System Engineering and Technology (ICSET). IEEE, 2019, pp. 172–177.
23. Parmenter, D. *Key performance indicators: developing, implementing, and using winning KPIs*, 3rd ed.; Wiley, 2015.
24. Penzenstadler, B. Towards a definition of sustainability in and for software engineering. In Proceedings of the Proceedings of the 28th Annual ACM Symposium on Applied Computing (SAC). ACM Press, 2013, p. 1183.
25. Venters, C.C.; Capilla, R.; Betz, S.; Penzenstadler, B.; Crick, T.; Crouch, S.; Nakagawa, E.Y.; Becker, C.; Carrillo, C. Software sustainability: Research and practice from a software architecture viewpoint. *Journal of Systems and Software* **2018**, *138*, 174–188.
26. International Organization for Standardization [ISO]. Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models. Technical Report ISO/IEC 25010:2011, 2011.
27. Lago, P. Architecture Design Decision Maps for Software Sustainability. In Proceedings of the International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS). IEEE/ACM, 2019, pp. 61–64.
28. Chaffey, D. OGSM model: A strategic framework for top-down vision realization. Available online: https://www.smartinsights.com/marketing-planning/marketing-models/ogsm-model-framework/ (accessed on 2024-01-30).
29. Darke, P.; Shanks, G.; Broadbent, M. Successfully Completing Case Study Research: Combining Rigour, Relevance and Pragmatism. *Information Systems Journal* **1998**, *8*, 273–289.
30. Runeson, P.; Höst, M. Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering - An International Journal* **2009**, *14*, 131–164.

31. Brereton, P.; Kitchenham, B.; Budgen, D.; Li, Z.H. Using a Protocol Template for Case Study Planning. In Proceedings of the International Conference on Evaluation and Assessment in Software Engineering (EASE). BCS Learning and Development Ltd, 2008.
32. Brown, S. The C4 Model for Software Architecture. Available online: https://c4model.com (accessed on 2021-07-08).
33. Müller, J.K. The building block method: component-based architectural design for large software-intensive product families. PhD Thesis, Universiteit van Amsterdam, 2003.
34. Kontio, J.; Bragge, J.; Lehtola, L. The Focus Group Method as an Empirical Tool in Software Engineering. In *Guide to Advanced Empirical Software Engineering*; Shull, F.; Singer, J.; Sjøberg, D.I.K., Eds.; Springer London, 2008; pp. 93–116.
35. Felderer, M.; Travassos, G.H., Eds. *Contemporary Empirical Methods in Software Engineering*; Springer, 2020.
36. Boehm, J.; Curcio, N.; Merrath, P.; Shenton, L.; Stähle, T. The risk-based approach to cybersecurity, 2020.
37. Splunk Inc. Available online: https://www.splunk.com (accessed on 2024-01-30).
38. IBM Maximo IT. Available online: https://www.ibm.com/products/maximo/it-asset-management (accessed on 2024-01-30).
39. Atlassian. Jira - Issue & Project Tracking Software. Available online: https://www.atlassian.com/software/jira (accessed on 2024-01-30).
40. Red Hat Inc. Official Red Hat OpenShift Documentation. Available online: https://docs.openshift.com (accessed on 2024-01-30).
41. Qualys. IT Security and Compliance Platform. Available online: https://www.qualys.com (accessed on 2024-01-30).
42. Andersen, B. *Business process improvement toolbox*, 2nd ed.; ASQ Quality Press, 2007.
43. Kaczynski, D.; Wood, L.; Harding, A. Using radar charts with qualitative evaluation: Techniques to assess change in blended learning. *Active Learning in Higher Education* **2008**, *9*, 23–41.
44. Patro, S.K.; sahu, K.K. Normalization: A Preprocessing Stage. *International Advanced Research Journal in Science, Engineering and Technology* **2015**, pp. 20–22.
45. Boudon, R. What middle-range theories are, 1991.
46. Wohlin, C.; Runeson, P.; Höst, M.; Ohlsson, M.C.; Regnell, B.; Wesslén, A. *Experimentation in Software Engineering*; Springer, 2012.
47. Hristov, I.; Chirico, A. The Role of Sustainability Key Performance Indicators (KPIs) in Implementing Sustainable Strategies. *Sustainability: Science Practice and Policy* **2019**, *11*.
48. Kaplan, R.S.; Norton, D.P. The Balanced Scorecard-Measures That Drive Performance. *Harvard Business Review* **1992**.
49. Kaplan, R.S.; Norton, D.P. Using the Balanced Scorecard as a Strategic Management System. *Harvard Business Review* **1996**.
50. Chai, N. *Sustainability Performance Evaluation System in Government: A Balanced Scorecard Approach Towards Sustainable Development*; Springer, 2009.
51. Verdecchia, R.; Lago, P.; Ebert, C.; De Vries, C. Green IT and Green Software. *IEEE Software* **2021**, *38*, 7–15.
52. Venters, C.C.; Jay, C.; Lau, L.; Griffiths, M.K.; Holmes, V.; Ward, R.R.; Austin, J.; Dibsdale, C.E.; Xu, J. Software sustainability: The modern tower of babel. In Proceedings of the CEUR Workshop Proceedings. CEUR, 2014, Vol. 1216, pp. 7–12.
53. Venters, C.C.; Kocak, S.A.; Betz, S.; Brooks, I.; Capilla, R.; Chitchyan, R.; Duboc, L.; Heldal, R.; Moreira, A.; Oyedeji, S.; et al. Software Sustainability: Beyond the Tower of Babel. In Proceedings of the International Workshop on Body of Knowledge for Software Sustainability (BoKSS). IEEE/ACM, 2021, pp. 3–4.
54. Groher, I.; Weinreich, R. An Interview Study on Sustainability Concerns in Software Development Projects. In Proceedings of the Euromicro Conference on Software Engineering and Advanced Applications (SEAA). IEEE, 2017, pp. 350–358.
55. Koziolek, H.; Domis, D.; Goldschmidt, T.; Vorst, P. Measuring Architecture Sustainability. *IEEE Software* **2013**, *30*, 54–62.
56. Capilla, R.; Nakagawa, E.Y.; Zdun, U.; Carrillo, C. Toward Architecture Knowledge Sustainability: Extending System Longevity. *IEEE Software* **2017**, *34*, 108–111.
57. Ojameruaye, B.; Bahsoon, R.; Duboc, L. Sustainability debt: a portfolio-based approach for evaluating sustainability requirements in architectures. In Proceedings of the International Conference on Software Engineering Companion. ACM, 2016, pp. 543–552.
58. Naumann, S.; Dick, M.; Kern, E.; Johann, T. The GREENSOFT Model: A reference model for green and sustainable software and its engineering. *Sustainable Computing: Informatics and Systems* **2011**, *1*, 294–304.
59. Kalaitzoglou, G.; Bruntink, M.; Visser, J. A Practical Model for Evaluating the Energy Efficiency of Software Applications. In Proceedings of the International Conference on ICT for Sustainability (ICT4S), 2014, pp. 77–86.
60. Koçak, S.A.; Alptekin, G.I.; Bener, A.B. Evaluation of Software Product Quality Attributes and Environmental Attributes using ANP Decision Framework. In Proceedings of the Third International Workshop on Requirement Engineering for Sustainability, 2014, pp. 37–44.
61. Manotas, I.; Bird, C.; Zhang, R.; Shepherd, D.; Jaspan, C.; Sadowski, C.; Pollock, L.; Clause, J. An empirical study of practitioners' perspectives on green software engineering. In Proceedings of the International Conference on Software Engineering. ACM, 2016, pp. 237–248.
62. Andrikopoulos, V.; Lago, P. Software Sustainability in the Age of Everything as a Service. In *Next-Gen Digital Services. A Retrospective and Roadmap for Service Computing of the Future*; Aiello, M.; Bouguettaya, A.; Tamburri, D.A.; van den Heuvel, W.J., Eds.; Springer, 2021; Vol. 12521, *LNCS*, pp. 35–47.
63. Saputri, T.R.D.; Lee, S.W. Software Analysis Method for Assessing Software Sustainability. *International Journal of Software Engineering and Knowledge Engineering* **2020**, *30*, 67–95.
64. Penzenstadler, B.; Raturi, A.; Richardson, D.; Tomlinson, B. Safety, Security, Now Sustainability: The Nonfunctional Requirement for the 21st Century. *IEEE Software* **2014**, *31*, 40–47.

65. Kalaitzoglou, G.; Bruntink, M.; Visser, J. A Practical Model for Evaluating the Energy Efficiency of Software Applications:. In Proceedings of the ICT for Sustainability 2014 (ICT4S-14), Stockholm, Sweden, 2014.
66. Mahmoud, S.S.; Ahmad, I. A Green Model for Sustainable Software Engineering. *International Journal of Software Engineering and Its Applications* **2013**, *7*, 20.
67. Verdecchia, R.; Lago, P.; Malavolta, I.; Ozkaya, I. ATDx: Building an Architectural Technical Debt Index. In Proceedings of the International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE). SciTePress, 2020, pp. 531–539.
68. Sehestedt, S.; Cheng, C.H.; Bouwers, E. Towards quantitative metrics for architecture models. In Proceedings of the First International Conference on Dependable and Secure Cloud Computing Architecture (DASCCA). ACM Press, 2014, pp. 1–4.
69. Verdecchia, R.; Kruchten, P.; Lago, P.; Malavolta, I. Building and evaluating a theory of architectural technical debt in software-intensive systems. *Journal of Systems and Software* **2021**, *176*, 110925.
70. International Organization for Standardization [ISO]. Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - Quality measurement framework. Technical Report ISO/IEC 25020:2019, 2019.
71. Khanjani, A.; Rahman, W.N.W.A.; Ghani, A.A.A.; Sultan, A.B.M. SaaS Quality of Service Attributes. *Journal of Applied Sciences* **2014**, *14*, 3613–3619.